

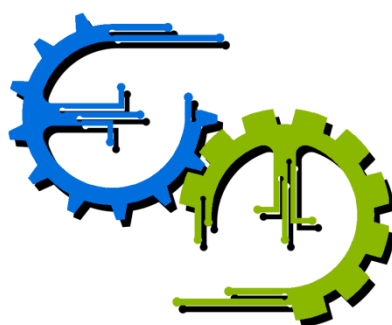


TRABALHO DE GRADUAÇÃO

**Reconhecimento de Gestos usando
Redes Neurais Artificiais**

Por,
Luan Caius Ramos

Brasília, Julho de 2011



**ENGENHARIA
MECATRÔNICA**
UNIVERSIDADE DE BRASÍLIA

UNIVERSIDADE DE BRASÍLIA
Faculdade de Tecnologia
Curso de Graduação em Engenharia de Controle e Automação

TRABALHO DE GRADUAÇÃO

Reconhecimento de Gestos usando Redes Neurais Artificiais

POR,

Luan Caius Ramos

Relatório submetido como requisito parcial para obtenção
do grau de Engenheiro de Controle e Automação.

Banca Examinadora

Prof. Dr. Alexandre Zaghetto, UnB/ CIC
(Orientador)

Prof. Dr. Bruno L. Macchiavello Espinoza ,
UnB/ CIC
(Examinador Interno)

Prof. Dr. Claudio Gottschalg Duque , UnB/ CID
(Examinador Externo)

Brasília, Julho de 2011

FICHA CATALOGRÁFICA

Luan, Caius Reconhecimento de Gestos usando Redes Neurais Artificiais, [Distrito Federal] 2011. xiv, 49p., 297 mm (FT/UnB, Engenheiro, Controle e Automação, 2011). Trabalho de Graduação – Universidade de Brasília.Faculdade de Tecnologia.	
1.Processamento de Imagem 3.Redes Neurais	2.Reconhecimento de Gestos 4.Inteligência Artificial
I. Mecatrônica/FT/UnB	

REFERÊNCIA BIBLIOGRÁFICA

Caius, Luan , (2011). Reconhecimento de Gestos usando Redes Neurais Artificiais. Trabalho de Graduação em Engenharia de Controle e Automação, Publicação FT.TG-nº 002/2011, Faculdade de Tecnologia, Universidade de Brasília, Brasília, DF, 49p.

CESSÃO DE DIREITOS

AUTOR: Luan Caius Ramos

TÍTULO DO TRABALHO DE GRADUAÇÃO: Reconhecimento de Gestos usando Redes Neurais Artificiais.

GRAU: Engenheiro

ANO: 2011

É concedida à Universidade de Brasília permissão para reproduzir cópias deste Trabalho de Graduação e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte desse Trabalho de Graduação pode ser reproduzida sem autorização por escrito do autor.

Luan Caius Ramos
SQN 209 Bloco C – Asa Norte.
70854-030 Brasília – DF – Brasil.

Dedicatória

À Elza, minha mãe, que me mostrou a importância do estudo através de inúmeros exemplos.

*Ao Lucian, meu irmão, que tornou menos cansativo e mais divertido todos os desafios que
encontrei na vida acadêmica.*

Agradecimentos

Agradeço a todos os meus amigos pelas dicas e contribuições nesse trabalho e ao professor Alexandre Zaghetto pela oportunidade de realizar esse projeto e por me ajudar em todos os desafios que apareceram.

RESUMO

Sistemas automáticos baseados em reconhecimento de gestos têm sido muito pesquisados para desenvolvimento de aplicações como interação homem-máquina, controle robótico e reconhecimento da linguagem de sinais. Este projeto tem como objetivo fazer reconhecimento de gestos de forma rápida e eficiente. Para isso foram utilizadas técnicas de processamento de imagem em conjunto com a classificação de redes neurais artificiais previamente treinadas. Com isso obteve-se uma aplicação, que usando uma simples webcam, consegue distinguir em um ambiente controlado até 10 gestos da ASL (American Sign Language). As aplicações possíveis a partir desse projeto vão desde reconhecimento completo da língua de sinais a até sistemas de segurança que utilizam gestos como codificação.

Palavras Chave: processamento de imagem, redes neurais, reconhecimento de gestos, inteligência artificial.

ABSTRACT

Automatic systems based on gesture recognition have been very researched for development of applications such as human-machine interaction, robotic control and recognition of sign language. This project aims to gesture recognition to quickly and efficiently. We used image processing techniques with the classification of artificial neural networks previously trained. This resulted in an application, that using a simple webcam, can distinguish in a controlled environment 10 different ASL gestures. Possible applications from this project go from full recognition of sign language to security systems that use gestures as a codification.

Keywords: image processing, neural networks, gesture recognition, artificial intelligence.

SUMÁRIO

1	INTRODUÇÃO	1
1.1	CONTEXTUALIZAÇÃO	1
1.2	DEFINIÇÃO DO PROBLEMA	1
1.3	OBJETIVOS DO PROJETO.....	2
1.4	APRESENTAÇÃO DO MANUSCRITO	3
2	PROCESSAMENTO DE IMAGENS DIGITAIS	4
2.1	INTRODUÇÃO	4
2.2	BREVE HISTÓRICO.....	5
2.3	ETAPAS FUNDAMENTAIS PARA UM SISTEMA DE PROCESSAMENTO DE IMAGEM.	5
2.4	REPRESENTAÇÃO DE IMAGENS.....	7
2.5	REPRESENTAÇÃO DE VÍDEOS	8
2.5.1	VIZINHANÇA E COMPONENTES CONEXOS	8
2.6	SISTEMA RGB	9
2.7	MORFOLOGIA MATEMÁTICA	10
2.7.1	DEFINIÇÕES BÁSICAS	10
2.7.2	DILATAÇÃO.....	11
2.7.3	EROSÃO.....	11
2.7.4	ABERTURA E FECHAMENTO.....	12
2.8	SEGMENTAÇÃO	12
2.8.1	DETECÇÃO DE DESCONTINUIDADE	13
2.8.2	LIMARIZAÇÃO	15
2.9	DESCRIÇÃO DE IMAGENS	15
2.9.1	ÁREA	16
2.9.2	PROJEÇÃO VERTICAL E HORIZONTAL	17
3	REDES NEURAIS ARTIFICIAIS	18
3.1	INTRODUÇÃO	18
3.2	CÉREBRO HUMANO.....	19
3.3	MODELOS DE NEURÔNIOS	19
3.3.1	MODELO BIOLÓGICO	19
3.3.2	MODELO DE McCULLOCH E PITTS.....	20
3.4	PERCEPTRON	22

3.5	REDE COM MÚLTIPLAS CAMADAS	23
3.6	APRENDIZADO	24
3.6.1	REGRA DELTA	25
3.6.2	BACKPROPAGATION	26
4	DESENVOLVIMENTO	28
4.1	INTRODUÇÃO	28
4.2	DETECÇÃO DA MÃO	28
4.3	EXTRAÇÃO DE CARACTERÍSTICAS PARA AS REDES NEURAIIS.....	30
4.4	TREINAMENTO E SIMULAÇÃO DAS REDES NEURAIIS.....	31
5	RESULTADOS EXPERIMENTAIS	34
5.1	INTRODUÇÃO	34
5.2	COMPILAÇÃO DOS RESULTADOS	34
5.3	ANÁLISE DOS RESULTADOS.....	35
6	CONCLUSÕES	38
	REFERÊNCIAS BIBLIOGRÁFICAS	39
	ANEXOS	41
I	TABELAS DE TESTES	42
I.1	TESTE 1	42
I.2	TESTE 2	42
I.3	TESTE 3	42
I.4	TESTE 4	42
I.5	TESTE 5	42
I.6	TESTE 6	42
I.7	PROJETO DA APLICAÇÃO NO SVN	42
II	DESCRIÇÃO DO CONTEÚDO DO CD	49

LISTA DE FIGURAS

1.1	Detecção de face com uma câmera	1
2.1	Exemplo de programa que reconhece expressões	4
2.2	Passos fundamentais em processamento de imagens	6
2.3	Convenção dos eixos para representação de imagens digitais	7
2.4	Exemplo de frames formando um vídeo	8
2.5	Tipos de vizinhança	8
2.6	Exemplo de rotulação de componentes conexos	9
2.7	Sistema de cores RGB	9
2.8	Exemplos de elementos estruturantes	11
2.9	Exemplos de Dilatação	11
2.10	Exemplos de Erosão	12
2.11	Exemplos de Abertura e Fechamento	13
2.12	Máscara genérica de 3x3 pixels	14
2.13	Máscara para detecção de Pontos	14
2.14	Exemplos de máscaras para detecção de retas	14
2.15	Exemplo de Binarização	15
2.16	Limiarização com conjunto específico de valores	16
2.17	Exemplo de projeções	17
3.1	Componentes de um Neurônio	20
3.2	Modelo do Neurônio de McCulloch e Pitts	20
3.3	Gráfico da função Sigmóide	21
3.4	Gráfico da função Signum	21
3.5	Gráfico da função Tangente Hiperbólica	21
3.6	Exemplo de um conjunto de classes	22
3.7	Modelo Perceptron com um neurônio	23
3.8	Exemplos de funções lógicas	23
3.9	Modelo de rede MLP	24
4.1	Segmentação Inicial	28
4.2	Operação de Fechamento	29
4.3	Remoção de pequenos blocos e buracos	29
4.4	Detecção da mão direita	30

4.5	Cálculo das projeções vertical e horizontal.....	31
4.6	Exemplo de entrada para as redes neurais	32
4.7	Ferramenta utilizada para treinamento das redes neurais	33
4.8	Gestos ASL usados para treinamento	33

LISTA DE TABELAS

3.1	Tabela verdade da função AND.....	23
3.2	Tabela verdade da função OR.....	23
3.3	Tabela verdade da função XOR.....	23
5.1	Comparação com 2 Neurônios.....	36
5.2	Comparação com 3 Neurônios.....	36
5.3	Comparação com 4 Neurônios.....	36
5.4	Comparação com 5 Neurônios.....	36
5.5	Comparação com 6 Neurônios.....	36
5.6	Comparação com 7 Neurônios.....	36
5.7	Comparação com 8 Neurônios.....	36
5.8	Comparação com 9 Neurônios.....	36
5.9	Comparação entre Conjuntos 1	37
5.10	Comparação entre Conjuntos 2	37
I.1	Teste 1 - 2 Neurônios	43
I.2	Teste 1 - 3 Neurônios	43
I.3	Teste 1 - 4 Neurônios	43
I.4	Teste 1 - 5 Neurônios	43
I.5	Teste 1 - 6 Neurônios	43
I.6	Teste 1 - 7 Neurônios	43
I.7	Teste 1 - 8 Neurônios	43
I.8	Teste 1 - 9 Neurônios	43
I.9	Teste 2 - 2 Neurônios	44
I.10	Teste 2 - 3 Neurônios	44
I.11	Teste 2 - 4 Neurônios	44
I.12	Teste 2 - 5 Neurônios	44
I.13	Teste 2 - 6 Neurônios	44
I.14	Teste 2 - 7 Neurônios	44
I.15	Teste 2 - 8 Neurônios	44
I.16	Teste 2 - 9 Neurônios	44
I.17	Teste 3 - 2 Neurônios	45
I.18	Teste 3 - 3 Neurônios	45
I.19	Teste 3 - 4 Neurônios	45

I.20	Teste 3 - 5 Neurônios	45
I.21	Teste 3 - 6 Neurônios	45
I.22	Teste 3 - 7 Neurônios	45
I.23	Teste 3 - 8 Neurônios	45
I.24	Teste 3 - 9 Neurônios	45
I.25	Teste 4 - 2 Neurônios	46
I.26	Teste 4 - 3 Neurônios	46
I.27	Teste 4 - 4 Neurônios	46
I.28	Teste 4 - 5 Neurônios	46
I.29	Teste 4 - 6 Neurônios	46
I.30	Teste 4 - 7 Neurônios	46
I.31	Teste 4 - 8 Neurônios	46
I.32	Teste 4 - 9 Neurônios	46
I.33	Teste 5 - 2 Neurônios	47
I.34	Teste 5 - 3 Neurônios	47
I.35	Teste 5 - 4 Neurônios	47
I.36	Teste 5 - 5 Neurônios	47
I.37	Teste 5 - 6 Neurônios	47
I.38	Teste 5 - 7 Neurônios	47
I.39	Teste 5 - 8 Neurônios	47
I.40	Teste 5 - 9 Neurônios	47
I.41	Teste 6 - 2 Neurônios	48
I.42	Teste 6 - 3 Neurônios	48
I.43	Teste 6 - 4 Neurônios	48
I.44	Teste 6 - 5 Neurônios	48
I.45	Teste 6 - 6 Neurônios	48
I.46	Teste 6 - 7 Neurônios	48
I.47	Teste 6 - 8 Neurônios	48
I.48	Teste 6 - 9 Neurônios	48

LISTA DE SÍMBOLOS

Símbolos Matemáticos

\mathbb{Z}	Conjunto dos Número inteiros
\mathbb{Z}^2	Espaço bidimensional inteiro
\mathbb{Z}^3	Espaço tridimensional inteiro
\in, \notin	Pertinência de um elemento
\oplus	Dilatação
Ω	Erosão
\cup	União entre conjuntos
\cap	Interseção entre conjuntos
\propto	Relação proporcional

Símbolos Gregos

α	Indica a inclinação de uma curva
Δ	Variação entre duas grandezas similares
ϕ	Função de Ativação de um neurônio
∂	Derivada parcial
δ	Gradiente local
η	Taxa de aprendizagem de uma rede neural

Sobrescritos

$'$	Derivada da função
$-$	Valor médio
\hat{A}	Reflexão do conjunto A
A^c	Complementar do conjunto A

Siglas

ABNT	Associação Brasileira de Normas Técnicas
ASL	American Sign Language
LIBRAS	Linguagem Brasileira de Sinais
SVN	Subversion
fps	Frames per second
pixel	Picture Element
RGB	Red Green Blue
RNA	Redes Neurais Artificiais
MCP	Modelo de McCulloch e Pitts
MLP	Multilayer perceptron
$V(40)$	Vetor de tamanho 40
$V(20)$	Vetor de tamanho 20

Capítulo 1

Introdução

1.1 Contextualização

O processamento de imagem tem sido muito utilizado nos últimos anos, com aplicações que variam desde detectar movimentos em sistemas de segurança até aplicações em jogos e video games. Um dos maiores desafios ainda consiste em fazer o computador entender o que significa cada elemento em uma certa imagem. Com a evolução dos microprocessadores, tal objetivo ficou mais viável e barato. Há cerca de 10 anos, um aplicativo usado para detectar rostos em uma imagem precisava de um computador poderoso para funcionar em tempo real, sendo que hoje uma simples câmera fotográfica possui tal recurso embarcado, como na Fig. 1.1. Portanto, produtos que usam processamento de imagem estão ficando mais baratos e sofisticados na sociedade. O controle de robôs usando apenas gestos está se tornando cada vez mais comum [1] e diversas empresas estão investindo muito nessa área devido à crescente demanda dos consumidores.



Figura 1.1: Detecção de face com uma câmera

1.2 Definição do problema

O reconhecimento de gestos é um passo fundamental para melhorar a interação homem máquina. Sem ele, um robô por exemplo, não conseguiria diferenciar os movimentos feitos por seu dono. Em uma sociedade que cada vez mais utiliza imagem como meio de transmissão de informa-

ção, ter um sistema que possa reconhecer gestos é um passo importante para a consolidação dos sistemas de visão computacional. Porém, também é necessário que tal reconhecimento de gestos seja feito de forma natural, ou seja, devem ser feitos gestos sem luvas ou marcações especiais. A não utilização de marcações especiais dificulta muito a identificação e classificação de um gesto. Uma mão normal sem luva, tem uma cor muito próxima do punho e do braço, e para reconhecer um gesto, é essencial delimitar a região de interesse para que um classificador possa ser usado. Se a região de interesse não for bem escolhida, é provável que o classificador falhe e portanto o sistema daria uma resposta errada. Uma das maiores dificuldades consiste justamente em fazer uma separação da região de interesse de forma robusta o suficiente para que na hora de tentar reconhecer se uma região é uma mão ou não, tenha-se uma resposta confiável na imagem. Diversas pesquisas surgiram utilizando técnicas diferentes de segmentação visando obter a cor de pele de forma rápida e precisa [2]. Outro grande desafio é utilizar os dados obtidos ao encontrar uma região que é mão, de forma a conseguir decifrar o gesto representado. Visando o reconhecimento do gesto, é necessário extrair da imagem descritores suficientes para que um gesto não seja confundido com outro. Gestos da *ASL* com formato parecidos são muito difíceis de serem reconhecidos. Porém, existem certos gestos que apresentam uma boa diferença entre eles. Foram esses o gestos escolhidos para reconhecimento pelo programa proposto.

1.3 Objetivos do projeto

Este projeto visa dois objetivos, que são interrelacionados, mas podem aprimorados independentemente:

- Reconhecer uma mão aberta em um ambiente controlado, dizendo se esta é a mão esquerda ou direita
- Realizar de forma rápida e robusta o reconhecimento de posturas manuais

Com esses objetivos principais concluídos, pode-se estender o projeto, visando novos objetivos como:

- Ampliar o número de gestos que podem ser reconhecidos.
- Fazer a portabilidade do projeto para um sistema móvel, como os smartphones, para torná-lo mais acessível e facilitar seu uso, pois não seria necessário utilizar um computador.
- Utilizar o reconhecimento de gestos para interação homem-máquina.
- Medir características específicas da mão, como cor, tamanho e formato, utilizando como identificador pessoal.
- Fazer o treinamento da rede na própria aplicação.

Isso exemplifica o escopo do projeto, mostrando qual abordagem foi enfatizada para alcançar os objetivos.

1.4 Apresentação do manuscrito

Nos capítulos 2 e 3 é feita uma revisão bibliográfica sobre cada um dos temas de estudo, que são processamento de imagens e redes neurais artificiais. Em seguida, o capítulo 4 descreve a metodologia empregada no desenvolvimento do projeto, descrevendo passo a passo o algoritmo e as técnicas utilizadas. Os resultados experimentais são discutidos no capítulo 5, seguido das conclusões no capítulo 6. Os anexos contém tabelas com resultados detalhados para cada tipo de rede simulada e o endereço SVN (Subversion) com o projeto completo da aplicação.

Capítulo 2

Processamento de Imagens Digitais

2.1 Introdução

A área de processamento de imagens têm crescido bastante devido à sua ampla variedade de aplicações. De acordo com [3], um dos maiores objetivos continua sendo fazer um computador entender o que está representado em uma imagem. Contudo, existem também diversos desafios em áreas como compressão e decodificação de imagem e vídeo e transmissão em tempo real vídeos. Conforme vão surgindo técnicas mais complexas, os sistemas que usam processamento de imagem passam a ter maior importância na sociedade. Há mais de uma década, produtos com um sistema de processamento de imagem embarcado eram raros. Hoje, temos desde aplicativos de segurança que detectam a presença de criminosos em aeroportos ou eventos públicos até sistemas que detectam as expressões feitas pelo rosto de uma pessoa, como demonstrado em [4], ilustrado pela Fig. 2.1.

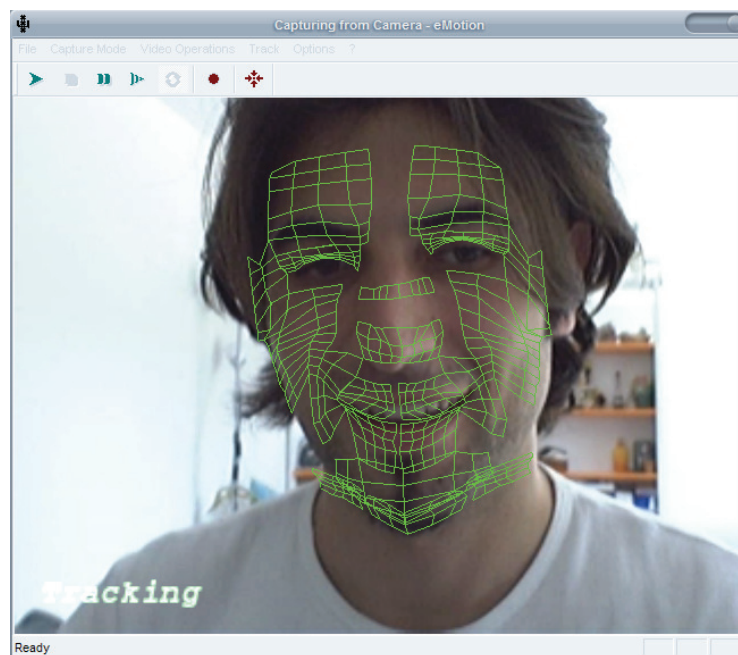


Figura 2.1: Exemplo de programa que reconhece expressões

Aplicações na área de automação estão usando cada vez mais câmeras para realizar seus propósitos. Tal demanda se justifica pela possibilidade de realizar tarefas cada vez mais complexas e muitas vezes perigosas com baixo custo devido ao crescente número de técnicas em processamento de imagem e ao aumento do poder computacional dos processadores utilizados pelos sistemas embarcados.

De acordo com trabalhos anteriores de [5], podemos estabelecer dois níveis de estudo em sistemas de visão computacional: processamento de imagem (baixo nível) e análise de imagens (alto nível). O processamento de imagens está relacionado a um conjunto de técnicas que podem alterar a imagem e extrair suas características, mas não interpretam aquilo que foi obtido. Entre essas técnicas, inclui-se aquisição de imagem, filtragem, segmentação, compressão, detecção de bordas, entre outros. A análise de imagens visa entender as características da imagem. Pode-se, por exemplo, fazer uma análise buscando comparar as texturas entre duas imagens diferentes ou encontrar uma pessoa em uma paisagem. Essa separação é apenas didática, pois na prática, geralmente, é necessário utilizar diversas técnicas para realizar um objetivo complexo. Para reconhecimento de gestos, por exemplo, pode-se utilizar uma segmentação para isolar o objeto de interesse e depois buscar as características do objeto para decidir se é ou não um gesto já conhecido.

2.2 Breve Histórico

Uma das primeiras aplicações utilizando imagens foi no melhoramento de imagens digitais enviadas por um cabo transatlântico, de Londres para New York, em meados de 1920 [3]. Desde então o maior foco da área de processamento de imagens era buscar otimizar a compressão de imagens para envio e melhorar imagens feitas por diversos aparelhos antigos. Na década de 60, boa parte das técnicas de processamento de imagem estavam sendo feitas por diferentes universidades e grupos nos Estados Unidos, com aplicações em áreas como reconhecimento de caracteres, imagens médicas, videoconferência, imagem por satélite, entre outros. Como o custo de pesquisa e operação era relativamente alto, os grupos eram restritos e com objetivos específicos. Com a popularização dos microcomputadores, diversas aplicações foram ganhando espaço nos centros de pesquisas pelo mundo, possibilitando um aumento significativo na quantidade de pesquisas. Hoje, é praticamente impossível encontrar uma fábrica de circuitos integrados que não tenha uma câmera de raios X para detecção de falhas nos seus produtos.

2.3 Etapas fundamentais para um sistema de Processamento de Imagem

Segundo Gonzalez [3], existem 5 passos que são fundamentais para executar uma tarefa de processamento de imagem, como mostra a Fig. 2.2.

1. O primeiro passo é o processo de aquisição da imagem, isto é, obter uma imagem digital. Geralmente esse processo é feito por um equipamento físico com sensores capazes de interpretar

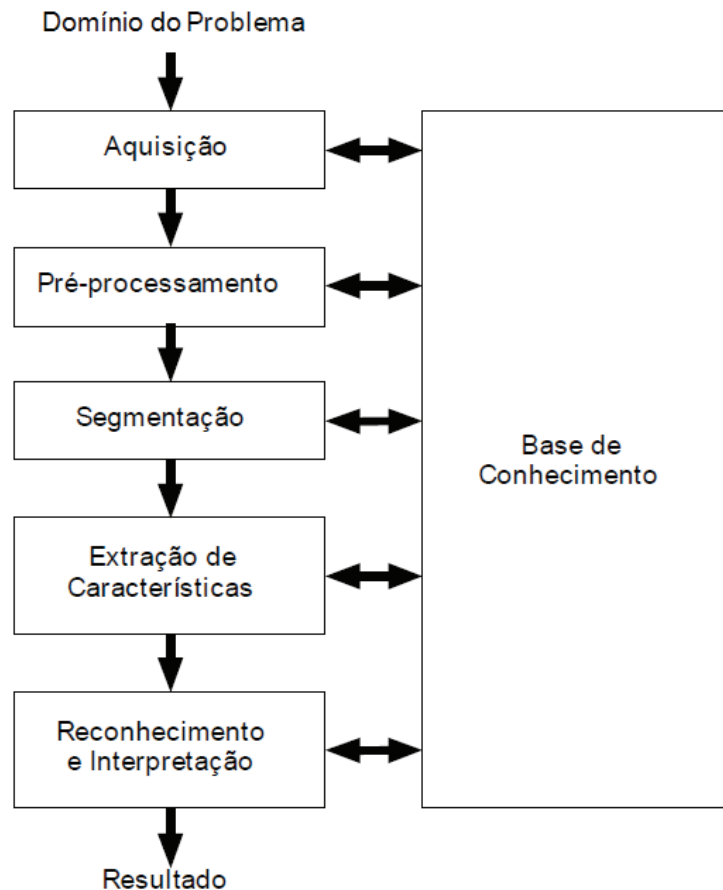


Figura 2.2: Passos fundamentais em processamento de imagens

o espectro de frequência da luz como um sinal analógico.

2. Depois de obter o sinal analógico da imagem, outros circuitos são responsáveis por transformar o sinal analógico em um sinal digital. Com esse sinal digital que se forma uma imagem. O pré-processamento consiste em fazer uma rápida modificação na imagem de forma a obter uma imagem melhor. Pode ser feito por exemplo, uma filtragem, o que eliminaria ruídos indesejáveis ou poderia realçar alguma característica.
3. No próximo passo temos a segmentação que é relativamente mais complexa que o pré-processamento e seu objetivo é encontrar uma região de interesse na imagem, podendo retornar uma imagem apenas com essa região.
4. Com a imagem já segmentada, é necessário criar uma representação para o que foi obtido. O processo de descrição, também chamado de seleção de características, busca extrair as características de interesse ou pelo menos diferenciar a imagem de outras.
5. Por último temos o reconhecimento e a interpretação, que dependendo da aplicação é o passo mais complexo. Seu resultado depende do objetivo visado. O reconhecimento busca separar os objetos da imagem em rótulos de forma que possam ser analisadas posteriormente. E a interpretação tenta dar um significado para o que reconhecido.

Todos os passos podem se comunicar com a base de conhecimento para conseguir realizar seus objetivos de forma mais confiável e rápida. Em um exemplo de detecção de faces, poderíamos ter uma imagem adquirida por uma câmera digital, que já realiza um pré-processamento na luminosidade da imagem. Em seguida essa imagem poderia ser segmentada para cor de pele e com o uso de alguma técnica para reconhecer faces com base nas distâncias entre os olhos e a boca, um rosto seria encontrado.

2.4 Representação de Imagens

Uma imagem é basicamente uma função bidimensional de intensidade da luz $f(x,y)$, em que x e y representam coordenadas espaciais e o valor de $f(x,y)$ corresponde ao nível de cinza (ou brilho) da imagem naquele ponto, conforme ilustra Fig. 2.3. Colocando cada elemento da função $f(x,y)$ em uma matriz de dimensão x por y , teremos uma imagem digital em que cada elemento dessa matriz é chamado de *pixel*. Portanto, toda a teoria de processamento de imagens usa a álgebra como base, pois suas operações ocorrem sempre em matrizes. Todas as operações envolvem basicamente a manipulação desses *pixels* em matrizes para obter uma nova matriz, ou seja, uma nova imagem.



Figura 2.3: Convenção dos eixos para representação de imagens digitais

2.5 Representação de Vídeos

Um vídeo é formado por um conjunto de imagens que ao ser mostrado sequencialmente cria a sensação de movimento. Geralmente essas imagens, ou *frames* são passados a uma taxa de 30 imagens por segundo, o que resulta em uma taxa de 30 *fps*. A Fig. 2.4 ilustra como quadros com pequenas diferenças formam um vídeo.

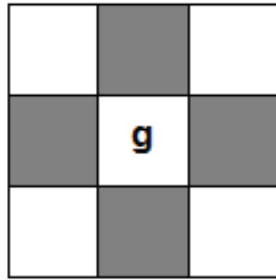


Figura 2.4: Exemplo de frames formando um vídeo

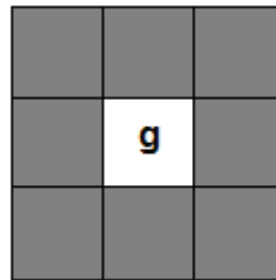
2.5.1 Vizinhaça e Componentes conexos

Um pixel g de coordenadas (x, y) possui dois vizinhos horizontais e dois verticais, cujas coordenadas são respectivamente $(x + 1, y)$, $(x - 1, y)$, $(x, y + 1)$ e $(x, y - 1)$. Esse pixels formam a chamada vizinhaça de 4 pixels, denotada por $N_4(g)$. Os quatro vizinhos diagonais de g são os pixels de coordenadas $(x - 1, y - 1)$, $(x - 1, y + 1)$, $(x + 1, y - 1)$ e $(x + 1, y + 1)$, constituindo o conjunto $N_d(g)$. A vizinhaça de 8 pixels, $N_8(g)$, é dada pela equação 2.1, conforme ilustra a Fig. 2.5(b).

$$N_8(g) = N_4(g) \cup N_d(g) \quad (2.1)$$



(a) Vizinhaça de 4 pixels



(b) Vizinhaça de 8 pixels

Figura 2.5: Tipos de vizinhaça

Um subconjunto de elementos L da imagem que são conexos entre si é chamado de *componente conexo*. Dois elementos f_1 e f_2 são conexos se existir um caminho de f_1 a f_2 contido em L . A Fig 2.6 mostra uma imagem bidimensional contendo três componentes conexos caso seja considerado a

vizinhança de 4 pixels ou dois componentes conexos caso seja considerado a vizinhança de 8 pixels.

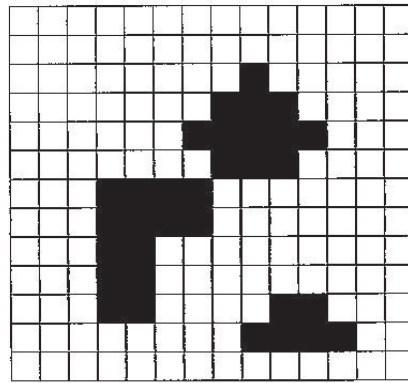


Figura 2.6: Exemplo de rotulação de componentes conexos

2.6 Sistema RGB

O modelo RGB (Red, Green, Blue), conforme a Fig. 2.7 é formado pela união das cores vermelha, verde e azul e está relacionado com a forma que as cores são percebidas na retina do olho, segundo [3]. RGB usa aditivos e mistura de cores e é o modelo mais usado na televisão, computadores, internet ou qualquer outro meio que projete cor com luz. Em uma imagem colorida, o sistema RGB é representado como a união dos luminosidade de cada cor do modelo. Portanto uma imagem coloridade pode ser representada por uma matriz tridimensional, contendo todas as uma matriz vermelha, uma verde e uma azul.



Figura 2.7: Sistema de cores RGB

2.7 Morfologia Matemática

A morfologia é uma área da biologia estuda a forma e estrutura de animais e plantas. Já a morfologia matemática é uma teoria formulada na década de 1960 e concentra-se na estrutura geométrica das imagens [5]. Ela utiliza a teoria dos conjuntos para representar a forma dos objetos em uma imagem e pode ser aplicada em diversas áreas de processamento de imagens, como realce, filtragem e segmentação. O princípio básico da morfologia matemática consiste em extrair as informações relativas à geometria e à topologia de um conjunto desconhecido (uma imagem), pela transformação através de outro conjunto completamente definido, chamado elemento estruturante. Por exemplo, o conjunto de todos os pixels pretos em uma imagem binária descreve completamente a imagem (uma vez que os demais pontos só podem ser brancos). Em imagens binárias, os conjuntos em questão são membros do espaço inteiro bidimensional \mathbb{Z}^2 , onde cada elemento do conjunto é um vetor 2-D cujas coordenadas são as coordenadas (x, y) do pixel preto (por convenção) na imagem. Imagens com mais níveis de cinza podem ser representadas por conjuntos cujos elementos estão no espaço \mathbb{Z}^3 . Neste caso, os vetores têm três elementos, sendo os dois primeiros as coordenadas do pixel e o terceiro seu nível de cinza.

2.7.1 Definições Básicas

Sejam A e B duas imagens binárias representados pelos conjuntos no espaço \mathbb{Z}^2 com componentes $a = (a_1, a_2)$ e $b = (b_1, b_2)$, respectivamente. Um conjunto A de coordenadas de pixels que satisfazem uma determinada condição pode ser expresso como $A = \{a \mid \text{condicao}\}$. A união de A e B , ou seja $A \cup B$, o conjunto dos elementos que pertencem a A , B ou ambos, expresso como

$$A \cup B = \{c \mid c \in A \text{ ou } c \in B\} \quad (2.2)$$

A intersecção de dois conjuntos A e B , denotada $A \cap B$, é o conjunto de todos os elementos que pertencem a ambos os conjuntos, dada por

$$A \cap B = \{c \mid c \in A \text{ e } c \in B\} \quad (2.3)$$

A translação de A pelo elemento p , denotada $A + p$, é definida como

$$A + p = \{a + p \mid a \in A\} \quad (2.4)$$

A reflexão de A , denotada \hat{A} , é definida como

$$\hat{A} = \{-a \mid a \in A\} \quad (2.5)$$

O complemento de A , denotado A^c , é o conjunto de todos os pixels que não pertencem a A , dado por

$$A^c = \{p \mid p \notin A\} \quad (2.6)$$

A diferença entre A e B , denotada $A - B$, é o conjunto de todos os elementos que pertencem a A , mas não pertencem a B , expressa por

$$A - B = \{p \mid p \in (A \cap B^c)\} \quad (2.7)$$

Um operador morfológico é um mapeamento entre o conjunto A que define a imagem e o conjunto B , chamado elemento estruturante, também definido em \mathbb{Z} . O elemento estruturante é expresso com respeito a uma origem local. Alguns exemplos de elementos estruturantes são mostrados na Fig. 2.8.

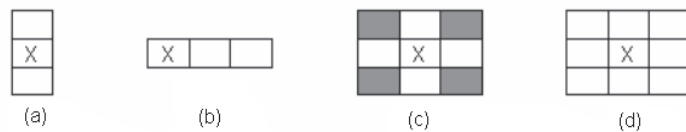


Figura 2.8: Exemplos de elementos estruturantes

2.7.2 Dilatação

A operação de dilatação entre o conjunto A e o elemento estruturante B é definida pela equação 2.8.

$$D(A, B) = A \oplus B = \{p \in \mathbb{Z}^2 \mid (\hat{B}_p \cap A) \neq \emptyset\} \quad (2.8)$$

O processo de dilatação entre A e B corresponde ao conjunto de todas as translações de B com os pontos da imagem em que há pelo menos um elemento não nulo (pixel com valor 1) em comum com o conjunto A . Para realizar a dilatação pode-se seguir os passos abaixo: Encontre a reflexão de \hat{B} do conjunto B transladando-o sobre sua origem. Se B é simétrico, esse passo não faz diferença. Translade \hat{B} deslocando p sobre A . $A \oplus B$ é o conjunto de todos os deslocamentos de p que mantém a intersecção de \hat{B}_p e A não vazio, ou seja, em contato. A dilatação de uma imagem binária com formato A por B expande o formato pela metade do tamanho de B , conforme mostrado nas Fig. 2.9(b).

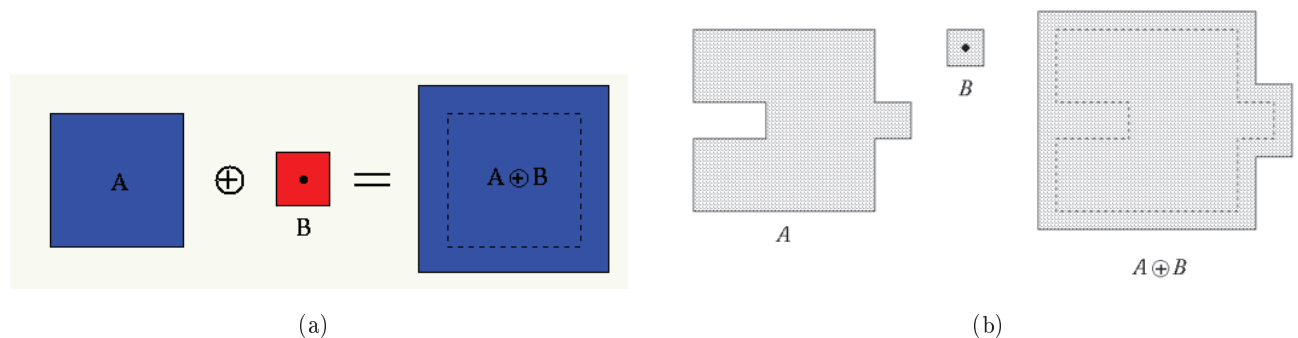


Figura 2.9: Exemplos de Dilatação

2.7.3 Erosão

A operação de erosão entre o conjunto A e o elemento estruturante B é definida pela equação 2.9.

$$E(A, B) = A \ominus B = \{p \in \mathbb{Z}^2 \mid B_p \subseteq A\} \quad (2.9)$$

A erosão de A por B corresponde ao conjunto de todos os elementos de B transladados por p que estão contidos em A . Entretanto deve-se observar que o resultado da erosão de uma imagem pode não ser um subconjunto da imagem original, caso o elemento estruturante não contenha a origem. $A \ominus B$ é o conjunto de todas as translações de p que mantêm a versão transladada de B totalmente contida em A . Erosão de uma imagem binária de formato A por B reduz o formato pela metade do tamanho de B , conforme mostrado na Fig. 2.10(a).

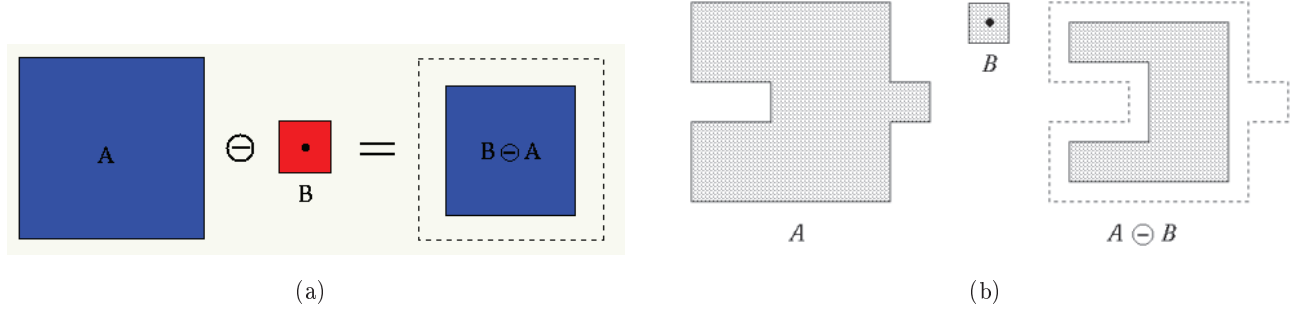


Figura 2.10: Exemplos de Erosão

2.7.4 Abertura e Fechamento

A abertura é uma operação composta, construída a partir de dois blocos básicos de operações. A abertura do conjunto A por B é encontrada ao fazer a erosão de A por B e em seguida a dilatação do resultado dessa operação por B [3].

$$O(A, B) = D(E(A, B), B), \quad A \circ B = (A \ominus B) \oplus B \quad (2.10)$$

Dilatação e erosão não são pares opostos de operações no sentido de que a erosão inicial não cancela a dilatação posterior. A erosão no início da abertura, elimina pequenos formatos (geralmente ruídos) e também encolhe o formato do objeto, enquanto a dilatação posterior faz o objeto expandir de volta, mas sem o ruído. O fechamento, como a abertura, também é formada por uma operação composta. O fechamento é feito com a dilatação de A por B e em seguida a erosão do resultado por B [3].

$$C(A, B) = E(D(A, B), B), \quad A \bullet B = (A \oplus B) \ominus B \quad (2.11)$$

Novamente, efeito da dilatação não cancela a erosão. A dilatação inicial elimina pequenos buracos no formato do objeto (geralmente ruídos) e expande o objeto, enquanto a erosão encolhe o objeto, porém sem os ruídos. Essas operações podem ser vistas na Fig. 2.11.

2.8 Segmentação

Como a interpretação dos dados contidos em imagens digitais é uma atividade complexa, um processo intermediário de segmentação é necessário para particionar o conjunto de dados de entrada em estruturas com conteúdo semântico relevante para a aplicação em questão. Essas estruturas correspondem a objetos ou partes de objetos que auxiliarão o processo de interpretação das imagens.

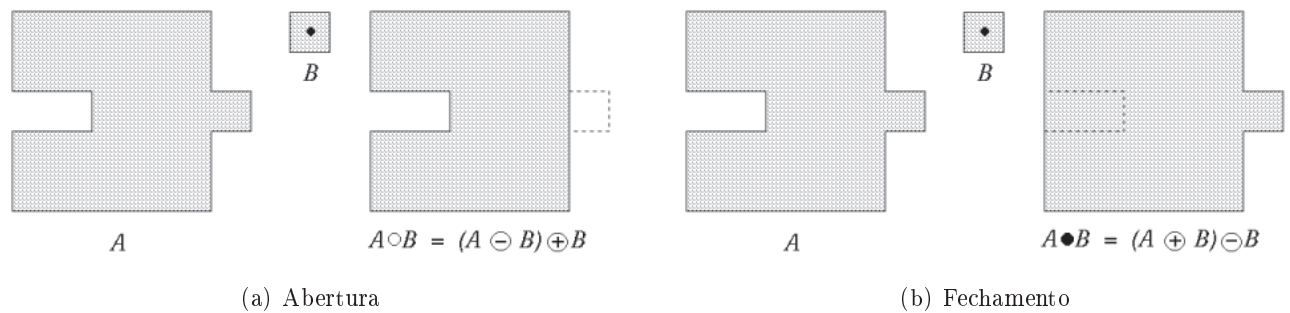


Figura 2.11: Exemplos de Abertura e Fechamento

Portanto, um processo de segmentação que identifique corretamente a localização, a topologia e a forma dos objetos é um requisito de fundamental importância para que as informações resultantes de um sistema de análise de imagens sejam confiáveis. Após a segmentação, cada objeto é descrito por meio de suas propriedades geométricas e topológicas. Por exemplo, atributos como área, forma e cor podem ser extraídos dos objetos e utilizados no processo de análise.

O processo de análise de imagens é considerado supervisionado quando há regiões da imagem em que se dispõe de informações que permitem a identificação de uma ou mais classes de interesses. Quando a região é associada a uma classe que é aprendida com base na similaridade entre amostras de treinamento, ou seja, são utilizadas técnicas para reconhecer as classes presentes na imagem, o processo de análise é chamado de não supervisionado.

As abordagens convencionais para segmentação de imagens são normalmente baseadas em imagens com níveis de cinza, buscando encontrar discontinuidades ou similaridades na imagem e podem ser divididas em 2 categorias principais. A primeira visa particionar a imagem ao encontrar variações abruptas nos níveis de cinza. A segunda categoria procura agrupar pontos da imagem que apresentam valores similares para um determinado conjunto de características, sendo que a seleção dessas características está relacionada com o domínio da aplicação e deve produzir um conjunto de regiões homogêneas.

2.8.1 Detecção de discontinuidade

Os tipos básicos de discontinuidades normalmente detectadas em imagens digitais são pontos, retas, junções e bordas. Uma maneira comum de identificação de discontinuidades é por meio da varredura da imagem por uma máscara, em um processo chamado convolução. No caso de uma máscara w com tamanho 3×3 , conforme mostrado em Fig. 2.12, esse procedimento envolve o cálculo da soma dos produtos dos coeficientes da máscara pelos níveis de cinza da região delimitada pela máscara. Dessa forma, a resposta R da máscara posicionada sobre um ponto da imagem é dada por:

$$R = w_1 z_1 + w_2 z_2 + \dots + w_9 z_9 = \sum_{i=1}^9 w_i z_i \quad (2.12)$$

em que z_i é o nível de cinza associado com o coeficiente w_i da máscara. A resposta da máscara é definida em relação à sua posição central. Quando a máscara é posicionada em um pixel da borda,

a resposta é calculada utilizando-se a vizinhança parcial.

$$w = \begin{array}{|c|c|c|} \hline w_1 & w_2 & w_3 \\ \hline w_4 & w_5 & w_6 \\ \hline w_7 & w_8 & w_9 \\ \hline \end{array}$$

Figura 2.12: Máscara genérica de 3x3 pixels

A partir de uma máscara 3x3 é possível fazer inúmeras detecções de discontinuidades. Por exemplo, pontos isolados em uma imagem pode ser encontrados fazendo-se a aplicação direta da máscara descrita pela Fig. 2.13.

-1	-1	-1
-1	8	-1
-1	-1	-1

Figura 2.13: Máscara para detecção de Pontos

Um ponto é detectado caso

$$|R| > T \quad (2.13)$$

em que T é um limiar não-negativo e R é dado pelo somatório da convolução, conforme a equação 2.12. Quando a máscara da Fig. 2.13 é posicionada sobre uma região homogênea da imagem, ou seja, todos os pixels da região possuem a mesma intensidade, a resposta da máscara é nula ($R = 0$).

Para encontrar retas, é utilizado o mesmo método da detecção de pontos, porém é necessário um tipo de máscara para cada orientação da reta, como pode ser visto pela Fig. 2.14. Essas máscaras são derivadas de segunda ordem, por isso detectam discontinuidades em uma imagem.

-1	-1	-1
2	2	2
-1	-1	-1

(a) Reta Horizontal

-1	-1	2
-1	2	-1
2	-1	-1

(b) Reta com 45°

-1	2	-1
-1	2	-1
-1	2	-1

(c) Reta Vertical

2	-1	-1
-1	2	-1
-1	-1	2

(d) Reta com 135°

Figura 2.14: Exemplos de máscaras para detecção de retas

2.8.2 Limiarização

A limiarização é uma das técnicas mais simples de segmentação e consiste na classificação dos pixels de uma imagem de acordo com a especificação de um ou mais limiares [3]. Uma maneira simples de separar um objeto do resto da imagem é selecionar um limiar T , de modo que para cada ponto (x, y) da imagem, é aplicada uma função $f(x, y)$ tal que se $f(x, y) > T$, então o ponto (x, y) , pertence ao objeto. Caso contrário, o ponto não pertence ao objeto e deve ser desconsiderado. Uma imagem limiarizada $g(x, y)$ pode ser definida como:

$$g(x, y) = \begin{cases} 1, & \text{se } f(x, y) > T \\ 0, & \text{se } f(x, y) \leq T \end{cases} \quad (2.14)$$

Portanto, os pixels rotulados com 1 correspondem ao objeto ou região de interesse, enquanto todos os pixels rotulados com 0 são desconsiderados e geralmente compõe o fundo da imagem. No caso da equação 2.14, ocorre uma binarização em que a imagem resultante possui apenas valores de intensidade, 0(preto) ou 1(branco). A Fig. 2.15 exemplifica uma binarização. A maior parte das aplicações existentes na área de processamento de imagem usam em algum ponto, uma binarização, pois a região de interesse fica em evidência, facilitando todas as operações subsequentes.



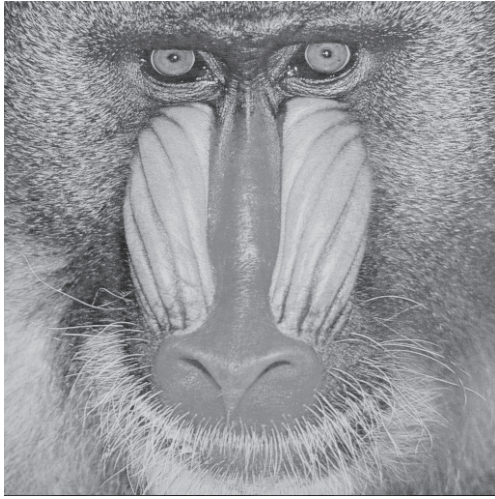
Figura 2.15: Exemplo de Binarização

Pode-se também utilizar limiarização para um conjunto específico de valores. Uma limiarização através de um conjunto de limiares é muito útil em imagens coloridas porque a cor definida em cada um de seus canais pode ser separada para outra imagem, obtendo assim uma segmentação por cor. Conforme a equação 2.15, uma imagem pode ter seus níveis convertido para um mesmo limiar, fazendo com que a imagem resultante tenha apenas os níveis definidos para cada intervalo, como mostrado na Fig. 2.16.

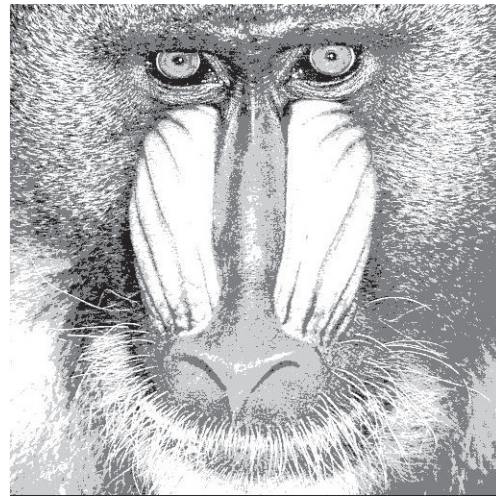
$$g(x, y) = \begin{cases} l_1, & \text{se } f(x, y) \leq T_1 \\ l_2, & \text{se } T_1 < f(x, y) \leq T_2 \\ l_3, & \text{se } f(x, y) > T_2 \end{cases} \quad (2.15)$$

2.9 Descrição de Imagens

Selecionar um conjunto de características extraídas de um objeto de interesse para poder classificá-lo é um dos problemas básicos no desenvolvimento de um sistema para análise de imagens. Com uma imagem segmentada em regiões ou objetos, os agrupamentos resultantes são normalmente representados ou descritos de acordo com o processamento subsequente.



(a) Imagem original



(b) Imagem com apenas 3 níveis de cinza

Figura 2.16: Limiarização com conjunto específico de valores

Um objeto pode ser representado em termos de suas características externas ou internas. A descrição depende da representação adotada e deve permitir a caracterização de sua forma sem ambiguidades e com um número reduzido de características ou medidas extraídas do objeto [5]. A forma do objeto é muito importante e tem sido utilizada como uma das características essenciais para descrever um objeto. Entretanto, em muitos casos, a forma do objeto pode sofrer distorções com a presença de ruídos ou degradações, o que prejudica sua classificação.

As técnicas de segmentação abordadas na seção anterior servem como base para o cálculo de descritores e para as representações utilizadas nesta seção. A união das técnicas de segmentação com as de descrição da imagem são necessárias nesse projeto para encontrar, treinar as redes neurais e reconhecer os gestos da mão.

2.9.1 Área

A área pode ser expressa como o número de pixels que compreende o objeto ou a região de interesse. Em uma imagem binária, a área de um objeto contido em um retângulo com dimensões $m \times n$ pixels, em que os pixels $f(x, y)$ do objeto são representados pelo valor 1, é definida por:

$$\text{Área} = \sum_{x=0}^{m-1} \sum_{y=0}^{n-1} f(x, y) \quad (2.16)$$

Dependendo da aplicação, o valor dado pela área é o tamanho real do objeto ou um valor proporcional. Embora a área seja um descritor muito importante, ela é normalmente usada para situações em que a escala do objeto não varia. A área é apenas invariante quanto à translação e rotação do objeto.

2.9.2 Projeção Vertical e Horizontal

Projeções Vertical e Horizontal podem ser usadas na imagem toda ou em uma região de interesse. A projeção vertical realiza o somatório de todos os pixels na vertical, ou seja, soma os pixels de cada coluna do objeto [5].

$$P_h(y) = \sum_{x=0}^{M-1} f(x, y) \quad (2.17)$$

A projeção horizontal faz a mesma operação para os pixels da horizontal, ou seja, soma cada linha do objeto.

$$P_v(x) = \sum_{y=0}^{N-1} f(x, y) \quad (2.18)$$

As Fig. 2.17(b) e (c) exemplificam o resultado do uso de projeções horizontal e vertical em uma imagem binarizada.

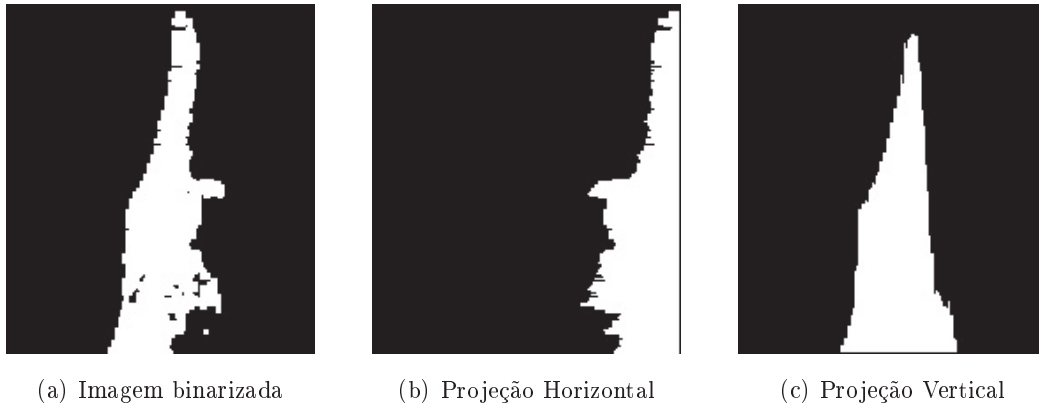


Figura 2.17: Exemplo de projeções

Capítulo 3

Redes Neurais Artificiais

3.1 Introdução

As RNAs (Redes Neurais Artificiais) são sistemas paralelos distribuídos compostos por unidades de processamento simples (nodos ou neurônios) que calculam determinadas funções matemáticas, podendo ser lineares ou não lineares [6]. Tais unidade são dispostas em uma ou mais camadas e interligadas por um grande número de conexões, geralmente unidirecionais. Na maioria dos modelos estas conexões estão associadas a pesos, os quais armazenam o conhecimento representado no modelo e servem para ponderar a entrada recebida por cada neurônio da rede. O funcionamento destas redes é inspirado em um estrutura física criada pela natureza: o cérebro humano.

O cérebro é bastante complexo, não linear e funciona de modo paralelo. Tem a capacidade de reorganizar suas estruturas, conhecidas como neurônios, para fazer operações de forma muito mais rápida e eficiente. O cérebro consegue fazer, por exemplo, reconhecimento de um rosto familiar em um ambiente desconhecido em cerca de 200 a 300 ms, enquanto tal operação pode levar horas em computadores.

A solução de problemas através de RNAs é muito atrativa, já que a forma como estes são representados internamente pela rede e o paralelismo natural inerente à arquitetura das RNAs criam a possibilidade de um desempenho superior ao dos modelos convencionais. Em RNAs, o procedimento usual da solução de problemas passa inicialmente por uma fase de aprendizagem, em que um conjunto de exemplos é apresentado para a rede, a qual extrai automaticamente as características necessárias para representar a informação fornecida. Estas características são utilizadas posteriormente para gerar resposta para o problema.

A capacidade de aprender através de exemplos e de generalizar a informação aprendida é o maior atrativo das RNAs para solução de problemas. A generalização, que está associada à capacidade da rede aprender através de um conjunto reduzido de exemplos e posteriormente dar respostas coerentes para dados não conhecidos, é uma demonstração de que as RNAs vão muito além do que simplesmente mapear relações de entrada e saída.

3.2 Cérebro Humano

O cérebro humano contém em torno de 10^{11} neurônios [6], sua célula fundamental. Cada um destes neurônios processa e se comunica com milhares de outros continuamente e em paralelo. A estrutura individual dos nodos, a topologia de suas conexões e o comportamento conjunto destes nodos naturais formam a base para o estudo das RNAs.

O cérebro humano é responsável pelos pensamentos, percepção, cognição, emoção e por tarefas motoras e sensoriais. Além disso, sua rede de nodos tem a capacidade de reconhecer padrões e relacioná-los, usar e armazenar conhecimento por experiência e também interpretar observações. Apesar do estudo aprofundado feito por diversos pesquisadores nas últimas décadas, o funcionamento das redes biológicas ainda não foi totalmente desvendado. Não se sabe ao certo a forma como as funções cerebrais são realizadas. Tem-se apenas modelos, que são atualizados a cada nova descoberta. Porém, a estrutura fisiológica básica destas redes é conhecida e nelas se baseiam as RNAs.

As RNAs tentam reproduzir as funções das redes biológicas, buscando implementar seu comportamento básico e sua dinâmica. E apesar de as redes artificiais se diferenciarem das redes biológicas no ponto de vista físico, o importante é observar as similaridades entre estes dois tipos de sistemas, tanto para que se possa entender melhor o sistema nervoso quanto para buscar idéias e inspirações para pesquisa em neurocomputação. Entre as características comuns, podemos citar o fato de serem sistemas baseados em unidades de computação paralela e distribuída, que possuem detectores de características, redundância e modularização das conexões. Essas características únicas permitem às RNAs reproduzir com fidelidade várias funções somente encontradas nos seres humanos.

3.3 Modelos de Neurônios

3.3.1 Modelo Biológico

O neurônio pode ser considerado análogo a uma unidade de processamento, pois aceita e combina estímulos de vários outros neurônios (muitas entradas), porém possui somente uma saída, como ilustra a Fig. 3.1. A intensidade dos estímulos que o neurônio recebe em suas entradas, denominadas dendritos, depende da proximidade física destas entradas com a saída, denominada axônio, de cada um dos neurônios que mandaram-lhe os estímulos. O soma, ou corpo do neurônio é capaz de funções mais complexas do que uma simples adição dos estímulos que recebeu pelos dendritos, porém considera-se um somatório simples como uma aproximação bem razoável. O axônio sempre aparece na saída do soma e pode ser considerado um dispositivo não-linear de disparo por produzir um pulso elétrico toda vez que o somatório dos sinais dentro do soma atinge um certo limiar crítico. Foi inspirando-se nesse modelo, que McCulloch e Pitts [7] em 1943 criaram o primeiro neurônio artificial.

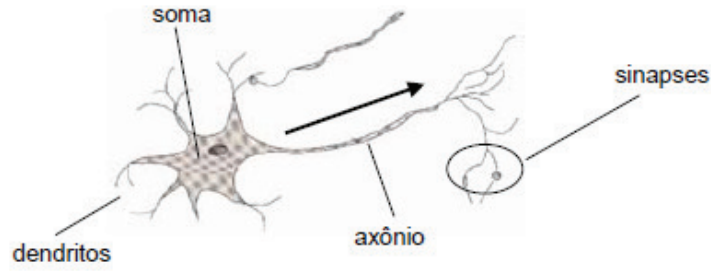


Figura 3.1: Componentes de um Neurônio

3.3.2 Modelo de McCulloch e Pitts

O modelo de McCulloch e Pitts [7], ou MCP, interpreta o funcionamento do neurônio como sendo um circuito binário simples que combina várias entradas e gera um sinal de saída. A Fig. 3.2 apresenta a lógica de um neurônio artificial seguindo esse modelo.

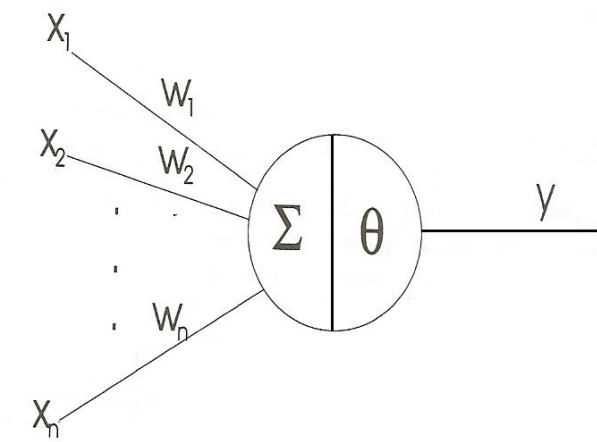


Figura 3.2: Modelo do Neurônio de McCulloch e Pitts

O neurônio do modelo de McCulloch terá sua saída ativa quando:

$$\sum_{i=0}^n x_i w_i \geq \theta \quad (3.1)$$

onde n é o número de entradas do neurônio, w_i é o peso associado à entrada x_i e θ é uma função de limiar do neurônio, conhecida como função de ativação. A função de ativação $f(\theta)$ usada no MCP era uma função degrau, dada pela fórmula:

$$f(\theta) = \begin{cases} 1, & \text{se } \theta > 0 \\ 0, & \text{se } \theta \leq 0 \end{cases} \quad (3.2)$$

Além da função degrau, pode-se utilizar outras funções como a sigmóide, signum e tangente hiperbólica.

Função Sigmóide: Ao contrário da função limiar, a função sigmóide pode obter todos os valores entre 0 e 1, de acordo com a equação 3.3 .

$$f(\theta) = \frac{1}{1 + e^{\alpha\theta}} \quad (3.3)$$

sendo α um parâmetro que quando tendendo ao infinito faz a função sigmóide se comportar como uma função degrau.

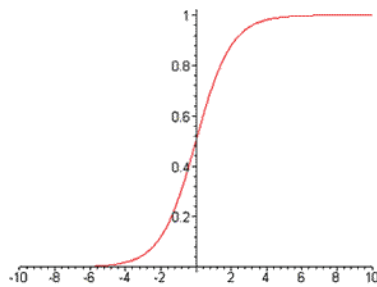


Figura 3.3: Gráfico da função Sigmóide

Função Signum: É uma função semelhante ao degrau, porém é limitada entre b e $-b$.

$$f(\theta) = b \frac{\theta}{|\theta|} \quad \theta \neq 0 \quad (3.4)$$

em que b representa os limites inferiores e superiores da função.

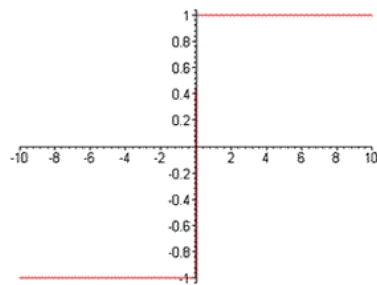


Figura 3.4: Gráfico da função Signum

Função Tangente Hiperbólica: Muito semelhante à função sigmóide, com valores entre $-b$ e b .

$$f(\theta) = a \frac{e^{(b\theta)} - e^{(-b\theta)}}{e^{(b\theta)} + e^{(b\theta)}} \quad (3.5)$$

sendo α o parâmetro de inclinação da curva e b os limites superiores e inferiores.

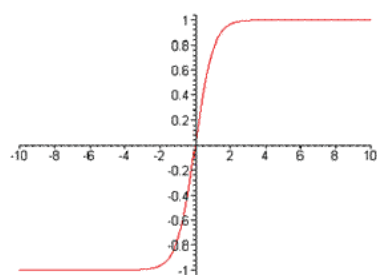


Figura 3.5: Gráfico da função Tangente Hiperbólica

McCulloch e Pitts simplificaram seu modelo considerando que os neurônios em cada camada da rede disparam com sincronismo, isto é, todos são avaliados ao mesmo tempo. Mas em sistemas biológicos, sabe-se que não existe um mecanismo para sincronizar as ações dos neurônios e o

valor da próxima saída de um neurônio depende essencialmente das ativações dos estados anteriores, já que até mesmo os neurotransmissores liberados anteriormente levam algum tempo para se recombinarem, influenciando assim as ativações seguintes.

O modelo MCP (Modelo de McCulloch e Pitts) possui as seguintes limitações:

- Redes MCP com apenas uma camada só conseguem implementar funções linearmente separáveis.
- Pesos negativos são mais adequados para representar disparos inibidores.
- O modelo foi proposto com pesos com pesos fixos, não-ajustáveis.

A partir do modelo de McCulloch e Pitts, surgiram várias outras abordagens usando diferentes funções de ativação, produzindo uma saída qualquer ou que não precisasse de uma entrada binária. Porém, esses novos modelos não tinham aprendizado. Foi somente com o trabalho de Frank Rosenblatt [8] que o conceito de aprendizagem foi introduzido.

3.4 Perceptron

Em 1958, Rosenblatt propôs o primeiro modelo para aprendizagem de RNAs de modo supervisionado, chamado Perceptron. O Perceptron é a forma mais simples que uma RNA pode ter para ser usada na classificação de padrões linearmente separáveis, ou seja, padrões que estão em lados opostos de um hiperplano, como mostra a Fig. 3.6.

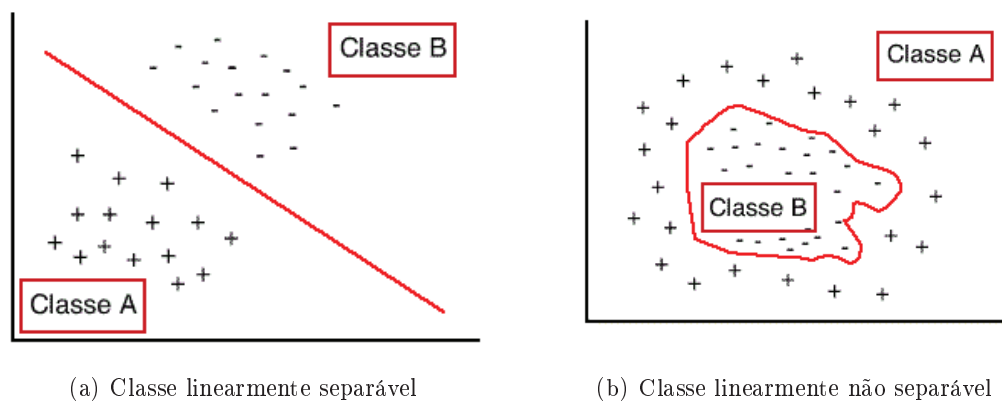


Figura 3.6: Exemplo de um conjunto de classes

Uma rede Perceptron é formada por um ou mais neurônios ligados com pesos sinápticos ajustáveis e uma polarização (*bias*), conforme Fig. 3.7.

O Perceptron consegue resolver problemas como a simulação de portas lógicas AND e OR, mas não consegue implementar a porta XOR. Para resolver problemas linearmente separáveis, o Perceptron usa um algoritmo de aprendizagem chamado Regra Delta [9], mostra que ela sempre converge. A prova de convergência do algoritmo é conhecida como Teorema de Convergência do

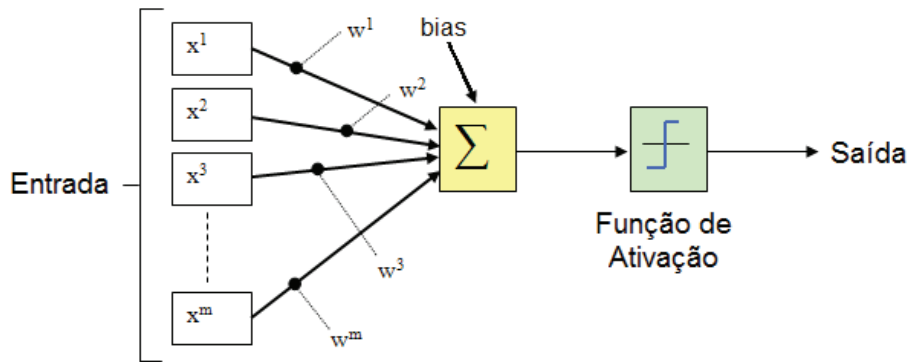


Figura 3.7: Modelo Perceptron com um neurônio

I_1	I_2	Saída
0	0	0
0	1	0
1	0	0
1	1	1

Tabela 3.1: Tabela verdade da função AND

I_1	I_2	Saída
0	0	0
0	1	1
1	0	1
1	1	1

Tabela 3.2: Tabela verdade da função OR

I_1	I_2	Saída
0	0	0
0	1	1
1	0	1
1	1	0

Tabela 3.3: Tabela verdade da função XOR

Perceptron e afirma que: se os padrões usados para treinar o Perceptron são retirados de duas classes linearmente separáveis, então o algoritmo Perceptron converge e posiciona a superfície de decisão na forma de um hiperplano entre as duas classes, conforme demonstrado na Fig. 3.8.

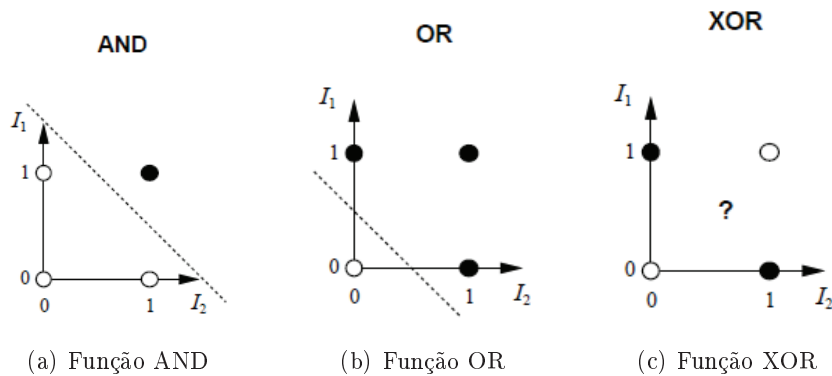


Figura 3.8: Exemplos de funções lógicas

3.5 Rede com Múltiplas Camadas

Como o perceptron não conseguia resolver problemas linearmente não separáveis, recebeu duras críticas de Minsky and Papert [10] e por cerca de uma década as pesquisas sobre redes neurais artificiais praticamente deixaram de existir. Somente no início dos anos 80 que o interesse voltou com a

descrição das redes de HopField [11] e do algoritmo de retrotransmissão de erro (*backpropagation*).

O perceptron de múltiplas camadas ou MLP, é uma rede com uma camada de entrada, que possui tantos nós de entrada quantos forem os sinais de entrada, uma ou mais camadas ocultas de neurônios e uma camada de saída com um número de neurônios igual ao número de sinais de saída. O sinal de entrada se propaga para frente através das camadas, ou seja, é uma rede alimentada para frente, conhecida como *feedforward*. Como pode ser visto na Fig. 3.9, o MLP é uma generalização da rede perceptron e também é treinado de forma supervisionada através de um regra de aprendizagem. O mecanismo utilizado pelo MLP para aprendizagem é o algoritmo de retrotransmissão de erro, também conhecido por *backpropagation*.

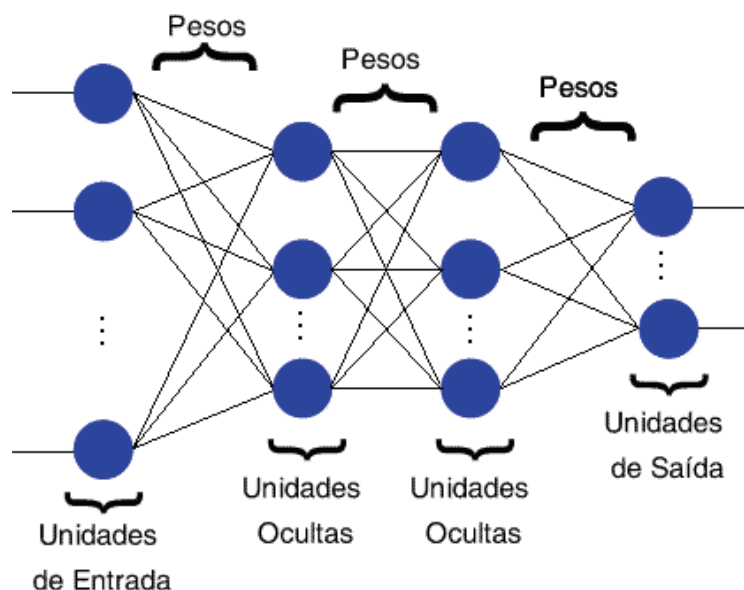


Figura 3.9: Modelo de rede MLP

3.6 Aprendizado

A utilização de uma RNA na solução de uma tarefa passa inicialmente por uma fase de aprendizagem, quando a rede extrai informações relevantes de padrões apresentados para ela, criando assim uma representação própria para o problema. A etapa de aprendizagem consiste em um processo iterativo de ajuste de pesos das conexões entre as unidades de processamento, que guardam ao final do processo, o conhecimento que a rede adquiriu do ambiente que está operando. Portanto o aprendizado de uma rede é específico para o treinamento dado nas entradas e na forma como os ajustes são realizados. Diversos métodos para treinamento de redes foram desenvolvidos, podendo ser agrupados em dois principais tipos: aprendizado supervisionado e aprendizado não-supervisionado.

Aprendizado não-supervisionado é um método que busca regularidades estatísticas nos dados de entrada. Portanto ele não possui conhecimento de qual deveria ser a saída, apenas recebe padrões de entrada. Apesar de parecer um aprendizado não natural e diferente dos seres humanos, ele aparece em diversos sistemas biológicos, como os estágios iniciais da visão e audição. Porém, esse

tipo de aprendizado só é possível se houver uma redundância nos dados de entrada, pois sem ela é impossível encontrar quaisquer padrões ou características. Conforme a rede vai recebendo os dados de entrada, sua estrutura interna tenta buscar uma representação codificada para as características dos dados de entrada. Existem diversos algoritmos de aprendizagem não supervisionada, como o aprendizado de Hebb, modelo de Linsker, modelo de Kohonen, regra de Oja e regra de Yuille, conforme apresentado em [6].

Aprendizado supervisionado é o método mais comum no treinamento de RNA e tem esse nome porque são fornecidas para a rede tanto a entrada quanto a saída desejada. O objetivo é encontrar uma relação entre os pares de entrada e saída. Conforme vão sendo apresentadas as entradas, a rede tem sua saída comparada com a saída desejada e com isso tem-se um erro da resposta atual. Esse erro causa o ajuste dos pesos das conexões de forma a minimizá-lo. Essa minimização é incremental de forma que pequenos ajustes nos pesos façam a rede ir em direção à solução do problema. Normalmente a soma dos erros quadráticos de todas as saídas é utilizada como parâmetro de medida do desempenho da rede e é utilizado pelos algoritmos de treinamento como função de custo a ser minimizada.

A maior desvantagem desse tipo de aprendizado é o fato de a rede não conseguir aprender exemplos que não foram passados no conjunto de treinamento. Os algoritmos mais conhecidos para essa forma de aprendizado são a regra Delta e sua generalização para múltiplas camadas, o algoritmo *backpropagation*.

3.6.1 Regra Delta

A regra delta é um algoritmo de aprendizado iterativo para o perceptron. Seu funcionamento segue os seguintes passos:

- Inicia-se os neurônios com pesos aleatório
- Apresenta-se um conjunto de entradas e calcula-se a resposta da rede
- Compara-se os valores calculados com os desejados, encontrando o erro
- Caso o erro não seja aceitável, ajusta-se o neurônio correspondente

O erro do neurônio é dado por:

$$E(j) = d(j) - y(j) \quad (3.6)$$

onde $E(j)$ é a diferença entre o sinal de saída desejado para o neurônio j , $d(j)$, e o sinal de saída calculado pela rede para aquele neurônio j , $y(j)$.

A forma de ajuste em cada neurônio é dado por:

$$w(i, j)_{T+1} = w(i, j)_T + \eta E(j)_T x(i) \quad (3.7)$$

sendo $w(i, j)_{T+1}$ o valor do peso corrigido, $w(i, j)_T$ o valor do peso na iteração anterior, i o índice de entrada, j o índice do neurônio, T a iteração, η a taxa de aprendizado e $x(i)$ o sinal de entrada.

O erro médio para todos os neurônios na iteração T será:

$$\varepsilon(T) = \frac{\sum_{j=1}^n |E(j)|}{n} \quad (3.8)$$

em que n é o total de neurônios da rede.

O erro médio para todo conjunto de treinamento será:

$$\varepsilon_m = \frac{\sum_{T=1}^n \varepsilon(T)}{n} \quad (3.9)$$

3.6.2 Backpropagation

A rede MLP, assim como o perceptron, necessita da definição do sinal de erro e da taxa de aprendizagem. Visando obter um melhor funcionamento e uma definição mais correta dos ajustes dos pesos das saídas dos neurônios em cada camada, são atribuídas algumas especificidades ao sinal de erro, o que garante uma maior convergência no aprendizado da rede. O sinal de erro do neurônio de saída j , na iteração n , pode ser definido conforme a equação 3.10:

$$e_j(n) = d_j(n) - y_j(n), \quad (3.10)$$

onde $y_j(n)$ é a resposta calculada para o neurônio j e $d_j(n)$ é a resposta desejada para o mesmo neurônio. O sinal de erro global da rede para uma iteração envolvendo os J neurônios da camada de saída é encontrado através de 3.11:

$$E(n) = \frac{1}{2} \sum_{j=1}^J e_j^2(n) \quad (3.11)$$

O erro de cada neurônio usa a distância euclidiana entre a resposta desejada e calculada pela rede para evitar que um erro de sinal negativo compense outro de sinal positivo.

O erro global médio da rede em um conjunto de treinamento é dado por:

$$\bar{E} = \frac{1}{N} \sum_{n=1}^N E(n), \quad (3.12)$$

em que N é o número de iterações necessárias para apresentar à rede todo o conjunto de treinamento, ou épocas.

A regra delta sugere que a variação dos pesos seja definida de acordo com o gradiente descendente do erro com relação ao peso, ou seja, que a variação do peso para um dado padrão siga a equação 3.13:

$$\Delta w_{ij} \propto \frac{\partial E}{\partial w_{ij}} \quad (3.13)$$

Conforme demonstrado em [12], após a derivada parcial $\partial E(n)/\partial w_{ij}(n)$ ser calculada, temos que:

$$\Delta w_{ij} = -\eta \delta_j(n) y_i(n), \quad (3.14)$$

sendo η a taxa de aprendizagem, y_i o vetor de sinais de saída de cada neurônio i , δ_j o gradiente local para o neurônio j definido por 3.15:

$$\delta_j(n) = -e_j(n)\phi'_j(v_j(n)), \quad (3.15)$$

onde $e_j(n)$ é o erro dado por 3.10, $v_j(n)$ é o somatório dos pesos com os valores de entrada mais o bias para o neurônio j e ϕ'_j é a derivada da função de ativação utilizada pelo neurônio j .

É importante notar que dado um dado peso w_{ij} , a saída do neurônio i é usada como base do cálculo ajuste do peso usando o gradiente local do neurônio j , conforme a equação 3.14. Esse processo se repete da saída para a entrada, ou seja, o neurônio que em uma iteração tinha sua saída usada no cálculo de ajuste do peso, terá em uma próxima iteração, seu gradiente local calculado e assim por diante até que todos os pesos e bias sejam ajustados. Portanto, o funcionamento do algoritmo backpropagation pode ser descrito em seis etapas:

1. Inicialização: os valores iniciais dos pesos e bias devem ser aleatórios, de preferência com uma distribuição uniforme e média igual a zero.
2. Apresentação dos dados de treinamento: apresenta-se uma época de exemplos à rede. Para cada exemplo será feita a propagação dos sinais e a correção dos pesos e níveis de bias utilizando o backpropagation, conforme descrito nos passos seguintes.
3. Propagação dos sinais: aplica-se o vetor de sinais de entrada na camada de entrada da rede e calcula-se o valor local de saída do neurônio e assim se prossegue pela camadas até chegar na saída. Com o vetor $y(n)$ obtido na saída, calcula-se o erro ao compará-lo com o vetor de saída desejado. Em seguida, são calculados o erro da iteração e o erro global.
4. Backpropagation: inicialmente calculam-se os gradientes locais para todos os neurônios da camada de saída usando a equação 3.15.

Depois são feitos ajustes dos pesos e bias através das equações 3.14 e 3.16.

$$\Delta b_j = -\eta\delta_j(n)y_i(n) \quad (3.16)$$

5. Retropropagação pelas camadas: aplica-se o algoritmo backpropagation à todas as camadas ocultas calculando-se o gradiente local através de 3.17 e seguindo as equações 3.18 e 3.19, em que y_k é o valor de saída do neurônio k da camada oculta.

$$\delta_j(n) = -e_j(n)\phi'_j(v_j(n)), \quad (3.17)$$

$$\Delta w_{ij} = -\eta\delta_j(n)y_k(n), \quad (3.18)$$

$$\Delta b_j = -\eta\delta_j(n)y_k(n) \quad (3.19)$$

6. Iteração: com isso conclui-se o treinamento de uma época. Para treinar a rede por completo com um certo número de épocas, deve-se escolher um critério de parada, que pode ser número máximo de iterações ou valor limite do erro global médio da rede.

Capítulo 4

Desenvolvimento

4.1 Introdução

Para se conseguir reconhecer gestos, foram necessários basicamente 3 passos, que estão divididos nas seções a seguir. Esses passos foram desenvolvidos ao longo do projeto, sendo que permaneceram os mesmos desde o começo, mudando apenas as técnicas envolvidas para cumprir seus objetivos. Em cada seção será mostrado o que foi feito e o resultado da abordagem utilizada.

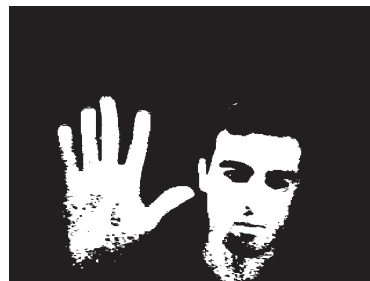
4.2 Detecção da mão

A tarefa inicial é encontrar a mão, porém dependendo das condições ambiente fica muito complexo realizar tal objetivo. Após a imagem ser adquirida por uma câmera de resolução 640x480 (Fig. 4.1(a)), é realizada uma segmentação por cor descrita em [2]. São utilizados os seguintes valores para a classificação como cor de pele cada canal RGB, obtendo a imagem da Fig. 4.1(b):

$$\begin{aligned} R &> 95 \text{ e } G > 40 \text{ e } B > 20 \text{ e} \\ \max(R,G,B) - \min(R,G,B) &> 15 \text{ e} \\ |R-G| &> 15 \text{ e } R > G \text{ e } R > B \end{aligned} \quad (4.1)$$



(a) Imagem Original



(b) Imagem Segmentada

Figura 4.1: Segmentação Inicial

Cada pixel da imagem é testado pelas condições descritas na equação 4.1 e somente se passar em todas elas que será considerado como pixel com cor de pele. Por usar o modelo RGB, essa segmentação depende muito da luminosidade da imagem, o que torna o processo menos robusto.

Após a segmentação estar concluída, teremos uma imagem binarizada. É feita uma operação de abertura, ou seja, uma erosão seguida de uma dilatação, visando retirar os ruídos da imagem, conforme ilustra a Fig. 4.2.



Figura 4.2: Operação de Fechamento

Com uma imagem binária sem ruídos, é feita uma rotulação de componentes conexos com uma filtragem por área, ou seja, componentes são rotulados e caso tenham uma área menor que a mínima especificada, serão desconsiderados nas análises seguintes. Foi utilizado como critério de área mínima o valor de 800, suficiente para remover pequenas regiões que não foram filtradas pela operação de fechamento. Após isso é realizado um preenchimento de regiões consideradas como *buracos*, que são regiões no interior de um componente conexo. A Fig. 4.3 mostra o resultado dessas duas operações na imagem.



Figura 4.3: Remoção de pequenos blocos e buracos

Temos agora apenas componentes conexos preenchidos e considerados grandes em relação à imagem. O próximo passo é buscar dentre cada componente conexo, aquele que é uma mão. Para isso foi utilizado como premissa que a mão a ser detectada tenha cinco dedos, esteja aberta e que seu dedo médio seja o dedo com maior altura. Se o dedo médio é o dedo mais alto, então a coordenada de altura inicial do componente conexo é ele próprio. Com isso é possível fazer uma busca horizontal na imagem para verificar se existe um dedo indicador e um dedo anelar.

Caso esses dedos existam e suas largura estejam na faixa de $\pm 30\%$ da largura do dedo médio, então considera-se que uma mão foi encontrada. Agora é necessário retirar algumas características básicas da mão, como sua altura e descobrir se ela é esquerda ou direita.

Ao fazer a segmentação, pode ocorrer de o punho, e até mesmo o braço, se juntarem com a mão formando um único grande componente conexo. Através da análise acima, a altura da mão iria do dedo médio até a parte mais baixa do componente conexo. Para evitar que isso ocorra, a altura da mão é definida como sendo 2.25 vezes a altura do dedo médio. Tal valor é suficiente para reduzir a altura da janela de busca da mão, de forma a enquadrá-la melhor e foi baseado em medidas empíricas do tamanho da mão de três indivíduos.

O último passo dessa abordagem é descobrir se a mão é esquerda ou direita. Ao fazer uma busca nas laterais da imagem de forma vertical, percebe-se que existe um padrão: caso a posição em y_1 encontrada na lateral esquerda seja maior que a posição y_2 encontrada na lateral direita, então essa é uma mão direita, conforme exemplificado pela Fig. 4.4. Caso y_1 seja menor que y_2 , então trata-se de uma mão esquerda.



Figura 4.4: Detecção da mão direita

Com isso a mão é enquadrada em uma janela menor para que as próximas operações utilizem menos memória e processamento.

4.3 Extração de características para as Redes Neurais

Através da janela obtida no passo anterior, são feitas as projeções vertical e horizontal. Com essas projeções distribuídas em seus respectivos vetores, é feita uma normalização dos valores utilizando os seguintes critérios:

- Defini-se um tamanho máximo fixo para um vetor, dividindo-o em intervalos.
- Cada intervalo corresponderá a um mesmo comprimento na projeção e deve ter também um valor máximo padrão
- O valor de cada intervalo será dado pelo somatório da projeção no intervalo escolhido vezes o valor máximo padrão dividido pelo tamanho da área do intervalo, conforme ilustra as

equações 4.2 e 4.3

$$P(i, j)_v = \frac{V_{max} * \sum_{x=i}^j f(x, y)}{\Delta I} \quad (4.2)$$

$$P(i, j)_h = \frac{V_{max} * \sum_{x=j}^i f(x, y)}{\Delta I} \quad (4.3)$$

onde V_{max} é o valor máximo padrão, $f(x, y)$ é um *pixel* da imagem bidimensional e ΔI é o tamanho do intervalo escolhido.

Assim é obtido um vetor de comprimento fixo que possui um valor máximo padrão. Um exemplo do processo de extração de características utilizando um vetor de tamanho (ΔI) 10 para as projeções e valor máximo padrão (V_{max}) de 11 pode ser visto na Fig. 4.5.

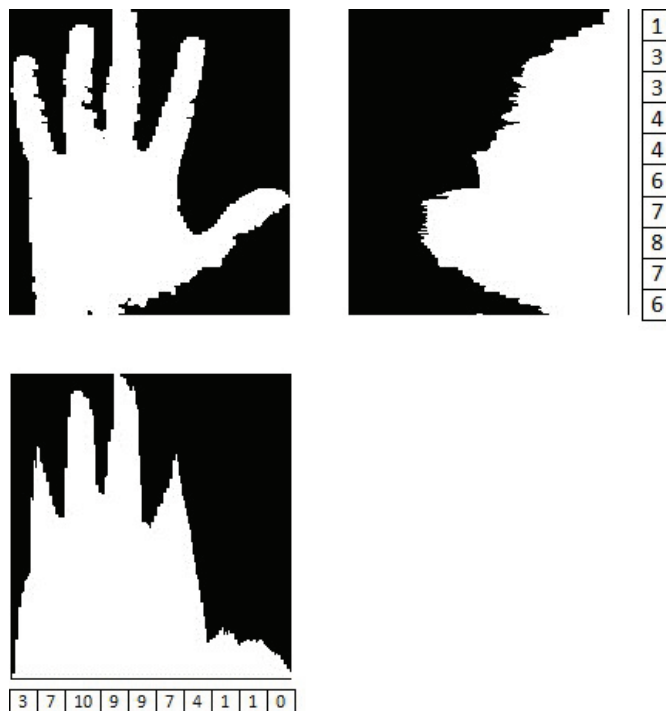


Figura 4.5: Cálculo das projeções vertical e horizontal

4.4 Treinamento e simulação das Redes Neurais

Com o vetor da projeção horizontal e o vetor da projeção vertical, é formado um novo vetor, constituído pela junção dos vetores das projeções normalizadas, ou seja, é necessário uní-los em um novo vetor e dividi-los pelo valor máximo padrão definido anteriormente para que sejam normalizados em um valor entre 0 e 1, como exemplifica a Fig. 4.6. Esse novo vetor formará a entrada da rede e será usado tanto para treinamento quanto para simulação.

O treinamento das redes neurais foi feito com o programa Matlab® (R2011a), através da ferramenta *nntraintool*. As especificações do treinamento podem ser vistas na Fig. 4.7, que mostra

1	3	3	4	4	6	7	8	7	6	3	7	10	9	9	7	4	1	1	0
---	---	---	---	---	---	---	---	---	---	---	---	----	---	---	---	---	---	---	---

(a) Vetor original

0,09	0,27	0,27	0,36	0,36	0,55	0,64	0,73	0,64	0,73	0,27	0,64	0,91	0,82	0,82	0,64	0,36	0,09	0,09	0,00
------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------

(b) Vetor normalizado

Figura 4.6: Exemplo de entrada para as redes neurais

uma rede com 20 entradas, 1 camada oculta com 10 neurônios e 2 saídas. O treinamento usa uma divisão de dados aleatória, ou seja, os pesos iniciais são inicializados de forma aleatória e o conjunto de treinamento é dividido igualmente em três partes: treinamento, validação e testes. O algoritmo de treinamento da rede é o Levenberg-Marquardt [13], que treina a rede de 10 a 100 vezes mais rápido que o método *backpropagation*. A performance da rede é medida de acordo com o erro global, dado por 3.12. A função de ativação usada foi a sigmóide, o que resultou em saídas com valores entre 0 e 1. Considerando N_1 o primeiro neurônio da saída e N_2 como o segundo, então uma saída será considerada positiva se $N_1 > N_2$ e negativa caso $N_1 < N_2$. Como os valores de N_1 e N_2 são próximos de 0 ou 1, não existe a possibilidade deles serem iguais. Logo, a rede responde com a saída 1 0 para uma entrada positiva e com a saída 0 1 para uma entrada negativa.

Todas as redes treinadas tinham de 20 a 40 vetores de entrada, de 2 a 10 neurônios e sempre 2 saídas. O treinamento ocorreu visando 10 gestos específicos da ASL. O gestos escolhidos para treinamento foram as letras A, B, C, D, H, I, L, V, W e Y, conforme mostra a Fig. 4.8

Para cada letra foram feitas duas divisões de treinamento, uma com vetores de tamanho 40 e outra com vetores de tamanho 20. Ao fazer a gravação dos vetores para treinamento, tentou-se variar ao máximo possível o gesto e a segmentação para obter um conjunto mais completo e abrangente, sendo que foram feitos conjuntos com 500, 1000 e 2000 vetores. Após obter os conjuntos de treinamento das redes, foram feitos conjuntos de simulação para cada tipo de rede, com 20 ou 40 entradas, totalizando 2000 valores para cada tipo. Com isso, treinou-se as redes, escolhendo a melhor entre 10 redes criadas para cada letra. O primeiro critério de escolha era a taxa de acerto para a letra e o segundo critério era a taxa de acerto se não fosse a letra. Caso uma entrada fosse a letra correspondente à rede, então a saída certa seria positiva, incrementando a taxa de acerto para a letra. E se uma entrada não fosse a letra da rede, então a saída certa seria negativa, incrementando a taxa de acerto quando não é a letra. Como foi feita uma rede neural para cada letra, utilizando 20 ou 40 vetores de entrada, com 500, 1000 ou 2000 conjuntos de treinamento e de 2 a 9 neurônio na camada oculta, obteve-se um total de 480 redes neurais.

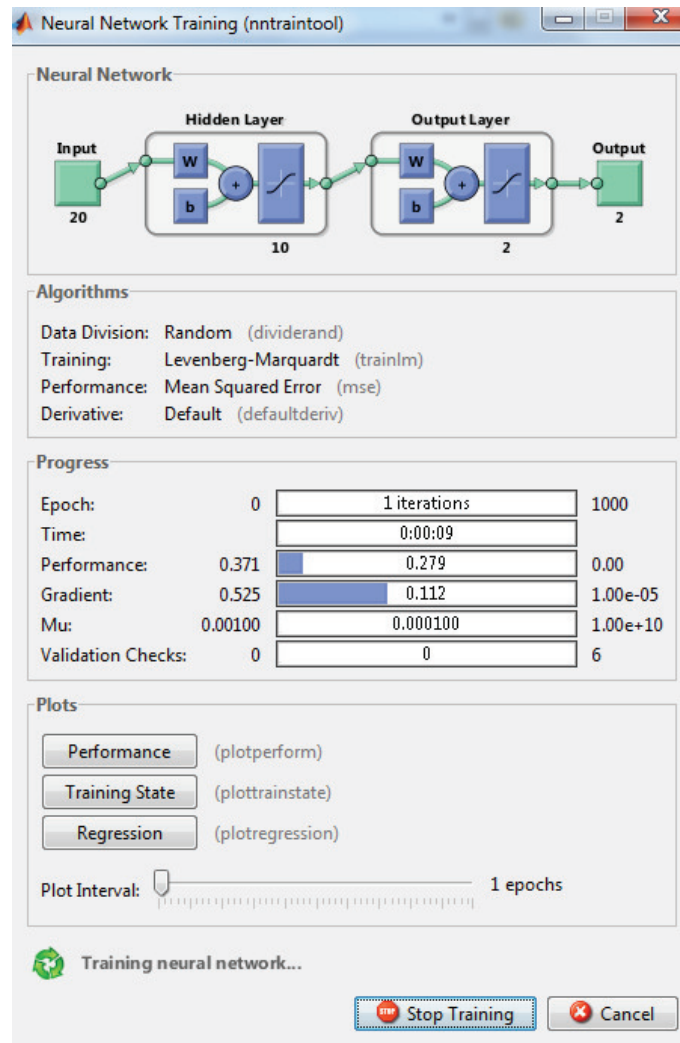


Figura 4.7: Ferramenta utilizada para treinamento das redes neurais

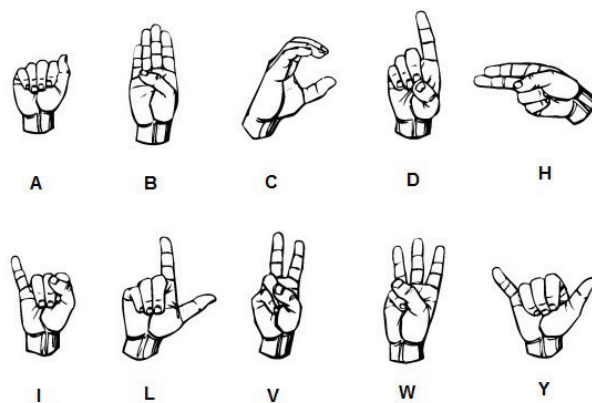


Figura 4.8: Gestos ASL usados para treinamento

Capítulo 5

Resultados Experimentais

5.1 Introdução

Os resultados obtidos dependem muito da segmentação inicial e todos os valores encontrados foram simulados utilizando a aplicação feita na linguagem C. A simulação da rede envolveu dois tipos diferentes de vetores para simulação: um vetor de tamanho 20 com 2000 valores e um vetor de tamanho 40 com 2000 valores. Cada um deles foi utilizado pra testar redes com o mesmo tamanho de entrada, sendo que os vetores usados para simulação não tiveram nenhuma interferência no treinamento.

5.2 Compilação dos Resultados

Foram obtidas 48 tabelas com os valores de 6 testes feitos de acordo com os seguintes critérios:

- Teste 1: Vetores de entrada de tamanho 20, $V(20)$, e conjunto de treinamento com 500 vetores
- Teste 2: Vetores de entrada de tamanho 20, $V(20)$, e conjunto de treinamento com 1000 vetores
- Teste 3: Vetores de entrada de tamanho 20, $V(20)$, e conjunto de treinamento com 2000 vetores
- Teste 4: Vetores de entrada de tamanho 40, $V(40)$, e conjunto de treinamento com 500 vetores
- Teste 5: Vetores de entrada de tamanho 40, $V(40)$, e conjunto de treinamento com 1000 vetores
- Teste 6: Vetores de entrada de tamanho 40, $V(40)$, e conjunto de treinamento com 2000 vetores

Os valores encontrados por todos esses testes estão nos anexos. Para facilitar a comparação entre os tipos de redes, foi feita uma compilação dos dados considerando o aumento da entrada da rede e o aumento no conjunto de treinamento. As tabelas de 5.1 a 5.8 ilustram a comparação entre uma rede com 20 entradas $V(20)$ e uma rede com 40 entradas $V(40)$.

As tabelas 5.9 e 5.10 comparam o aumento do conjunto de treinamento (épocas) para uma mesma rede.

5.3 Análise dos Resultados

Observa-se pelos resultados obtidos que a taxa de acerto das redes depende muito pouco do número de neurônios da camada oculta. O aumento da taxa de acerto ocorre quando a quantidade de valores de entrada da rede cresce (20 para 40) ou quando o conjunto de treinamento é maior (500 para 2000). A utilização de um maior número de entradas ajuda a rede a criar mais classes e assim conseguir separar melhor as letras, enquanto que aumentar o conjunto de treinamento faz a rede neural separar com maior certeza as diferentes classes testadas, mantendo seus pesos ajustados da melhor maneira possível.

As redes neurais foram treinadas para identificar se tal letra é ou não representada pelo vetor de entrada. Por ser uma abordagem simples, aumentar a quantidade de neurônios não reflete em maior taxa de acertos, pois o problema pode ser considerado de baixa complexidade. Mas caso fosse feita uma outra abordagem, como por exemplo, ter apenas uma única rede neural com 4 saídas e testar todos os conjuntos de dados nela, provavelmente seria necessário aumentar o número de neurônios e até mesmo o número de camadas ocultas.

Como o tempo de resposta das redes neurais é muito curto, uma forma de aumentar muito a taxa de acertos é colocar um maior número de entradas, que no caso poderia ser de 20 para 60 e também dobrar o conjunto de treinamento. Porém, tal abordagem somente teria sucesso se o incremento do número de entradas viesse acompanhado com uma maior variação dos vetores, o que facilitaria a separação em classes. Neste projeto não é relevante incrementar o número de entradas porque a extração de características continuaria a mesma, logo um vetor com 60 valores seria muito próximo de um vetor com 40 valores. O resultado seria um pequeno crescimento da taxa de acertos, mas um decréscimo significativo do tempo de resposta do sistema. A outra opção seria aumentar o conjunto de treinamento, de forma que seus vetores consigam abranger a maior parte das entradas possíveis da rede. Essa abordagem também pode aumentar a taxa de acertos, mas corre o risco de aumentar exponencialmente o tempo de treinamento da rede. A solução ideal seria conseguir extrair mais características da imagem e com isso montar um conjunto de dados mais abrangente para usá-lo no treinamento da rede neural.

Tabela 5.1: Comparação com 2 Neurônios

Total de Neurônios: 2		
Letra	Acerto com V(20)(%)	Acerto com V(40) (%)
A	53,08	75,67
B	82,08	91,77
C	94,58	98,87
D	33,65	64,75
H	92,25	88
I	82,68	75,62
L	95,95	85,03
V	77,92	77,28
W	63,05	58,62
Y	97,58	87,7
Média	77,28	80,33

Tabela 5.2: Comparação com 3 Neurônios

Total de Neurônios: 3		
Letra	Acerto com V(20)(%)	Acerto com V(40) (%)
A	51,18	84,37
B	89,98	99,75
C	95,52	96,92
D	33,37	60,7
H	93,05	88,72
I	76,42	71,78
L	93,57	86,43
V	83,47	83,97
W	59,75	61,63
Y	95,03	87,72
Média	77,13	82,2

Tabela 5.3: Comparação com 4 Neurônios

Total de Neurônios: 4		
Letra	Acerto com V(20)(%)	Acerto com V(40) (%)
A	54,42	80,85
B	88,28	96,95
C	70,75	99,3
D	46,3	61,1
H	92	80,07
I	76,48	78,1
L	88,12	86,53
V	76,95	72,37
W	61,02	62,75
Y	99,77	87,37
Média	75,41	80,54

Tabela 5.4: Comparação com 5 Neurônios

Total de Neurônios: 5		
Letra	Acerto com V(20)(%)	Acerto com V(40) (%)
A	48,4	78,38
B	90,2	95,83
C	97,65	99,25
D	48,52	60,15
H	94,45	86,3
I	82,32	78,58
L	89,37	81,18
V	88,93	75,2
W	57,62	63,32
Y	99,82	89,45
Média	79,73	80,77

Tabela 5.5: Comparação com 6 Neurônios

Total de Neurônios: 6		
Letra	Acerto com V(20)(%)	Acerto com V(40) (%)
A	49,9	88,78
B	73,92	98,6
C	96,68	99,72
D	52,22	68,68
H	92,12	82,38
I	88,55	87,1
L	85,38	90,55
V	75,42	79,63
W	69,5	68,75
Y	97,58	87,3
Média	78,13	85,15

Tabela 5.6: Comparação com 7 Neurônios

Total de Neurônios: 7		
Letra	Acerto com V(20)(%)	Acerto com V(40) (%)
A	43,2	89,28
B	88,2	98,8
C	95,4	99,27
D	47,07	71,2
H	92,98	82,78
I	85,92	80,03
L	93,07	85,93
V	87,55	80,83
W	59,23	64,78
Y	99,72	90,88
Média	79,23	84,38

Tabela 5.7: Comparação com 8 Neurônios

Total de Neurônios: 8		
Letra	Acerto com V(20)(%)	Acerto com V(40) (%)
A	51,4	87,2
B	88,33	97,7
C	96,28	98,5
D	41,65	71,18
H	88,83	82,15
I	82,35	69,02
L	78,05	83,1
V	88,43	81,58
W	66,02	62,52
Y	99,92	92,65
Média	78,13	82,56

Tabela 5.8: Comparação com 9 Neurônios

Total de Neurônios: 9		
Letra	Acerto com V(20)(%)	Acerto com V(40) (%)
A	51,25	97,05
B	76,63	98,6
C	98,07	98,07
D	54,88	64,03
H	95,07	83,82
I	88,05	73,95
L	92,75	82,78
V	73,55	77,37
W	59,1	75,93
Y	99,92	90,6
Média	78,93	84,22

Tabela 5.9: Comparação entre Conjuntos 1

Tamanho da entrada:20			
Letra	Conjunto 500 (%)	Conjunto 1000 (%)	Conjunto 2000 (%)
A	52,52	35,23	62,80
B	77,23	83,09	95,21
C	82,77	99,03	99,09
D	50,18	21,05	63,39
H	89,95	91,37	96,37
I	90,62	72,43	86,36
L	85,33	95,18	88,34
V	72,08	75,39	93,81
W	75,49	54,29	56,50
Y	97,90	99,86	98,65
Média	77,41	72,69	84,05

Tabela 5.10: Comparação entre Conjuntos 2

Tamanho da entrada:40			
Letra	Conjunto 500 (%)	Conjunto 1000 (%)	Conjunto 2000 (%)
A	92,49	75,32	87,74
B	95,53	97,03	99,79
C	99,33	99,63	96,54
D	51,79	77,67	68,40
H	82,94	89,51	80,10
I	78,72	60,69	91,63
L	75,89	81,51	97,62
V	89,02	70,88	72,72
W	36,17	87,25	73,34
Y	88,71	85,31	93,19
Média	79,06	82,48	86,11

Capítulo 6

Conclusões

O reconhecimento de gestos em tempo real sem a utilização de marcas ou luvas especiais, mostrou-se um grande desafio devido às diferentes técnicas necessárias para fazê-lo funcionar. Somente com uma perfeita integração de todas elas, pode-se realizar tal operação de forma confiável. A parte de maior relevância do sistema é a extração das características da mão, pois quanto maior a quantidade de dados, melhor ficaria o treinamento das redes neurais ou de qualquer outra técnica de inteligência artificial. Porém a abordagem adotada não conseguiu retirar dados suficientes para poder fazer uma identificação confiável dos 10 gestos escolhidos, apesar de conseguir em um ambiente controlado, acertar em média cerca de 80% dos gestos feitos.

A simulação e treinamento das redes neurais mostrou-se robusta e com uma quantidade abrangente de dados era possível fazer o reconhecimento de gestos mais simples de forma rápida e com baixo custo computacional. Portanto, a portabilidade de sistemas que utilizam redes neurais para aparelhos móveis é possível, desde que as redes sejam treinadas externamente. O treinamento é a parte que demanda mais tempo e dificilmente pode ser otimizada, já que é necessário que a rede encontre o caminho ótimo para a minimizar o erro de cada entrada.

Uma mudança na forma como é feita a segmentação, de modo a obter uma imagem da mão tendo pouca influência do ambiente, representaria um grande avanço e provavelmente o sistema seria capaz de identificar uma quantidade maior de gestos. Um exemplo de técnica atualmente utilizada para fazer a segmentação é a utilização de uma câmera infravermelha em conjunto com a câmera normal, conseguindo através de algoritmos complexos, uma maior confiabilidade do sistema.

Novas pesquisas envolvendo o reconhecimento do corpo e a predição de seus movimentos, como em [14], surgem com o auxílio de técnicas de rastreamento e com uso de inteligência artificial. Em relação às pesquisas feitas por grandes empresas, o reconhecimento de gestos é visto como uma especialidade do reconhecimento total do corpo e sua movimentação. Com isso, a interação homem-máquina fica cada vez mais próxima de se tornar uma realidade cotidiana devido às essas novas abordagens.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] THEIS, I. I. C.; STEINHAGE, A. Image processing methods for interactive robot control. *10th IEEE International Workshop*, p. 424–429, 2001.
- [2] VEZHNEVETS, V. S. V.; ANDREEVA, A. A survey on pixel-based skin color detection techniques. *Graphics and Media Laboratory*, 2001.
- [3] GONZALEZ, R. C.; WOODS, R. E. *Processamento de Imagens Digitais*. First. [S.l.]: EDGAR BLUCHER, 2000.
- [4] VALENTI, R.; SEBE, N.; GEVERS, T. Facial expression recognition: A fully integrated approach. In: *Workshop on Visual and Multimedia Digital Libraries*. [s.n.], 2007. Disponível em: <<http://www.science.uva.nl/research/publications/2007/ValentiVMDL2007>>.
- [5] PEDRINI, H.; SCHWARTZ, W. R. *Análise de imagens digitais: princípios, algoritmos e aplicações*. First. [S.l.]: Thomson Learning, 2008.
- [6] BRAGA, T. B. L. Antônio de P.; CARVALHO, A. C. P. de L. F. *Redes Neurais Artificiais, Teoria e aplicações*. First. [S.l.]: LTC, 2000.
- [7] MCCULLOCH, W.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, v. 5, p. 115–133, 1943.
- [8] ROSENBLATT, F. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychol. Rev.*, v. 65, p. 386–408, 1958.
- [9] ROSENBLATT, F. *Principles of Neurodynamics: Perceptrons and the theory of brain mechanisms*. first. [S.l.]: Spartan Books, 1962.
- [10] MINSKY, M.; PAPERT, S. Perceptrons: an introduction to computational geometry. *MIT Press*, 1969.
- [11] HOPFIELD, J. J. Neural networks and physical system with emergent collective properties. *Proc. Nat. Acad. Sci.*, 1982.
- [12] JR., O. L.; MONTGOMERY, E. *Redes Neurais, Fundamentos e Aplicações com Programas em C*. Ciência moderna. [S.l.]: Livros Técnicos e Científicos Editora, 2007.
- [13] HAGAN, M. T.; MENHAJ, M. Training feedforward networks with the marquardt algorithm. *IEEE Transactions on Neural Networks*, v. 5, p. 989–993, 1994.

- [14] SHOTTON ANDREW FITZGIBBON, M. C. T. S. M. F. R. M. A. K. A. B. J. Real-time human pose recognition in parts from single depth images. *IEEE*, 2011.

ANEXOS

I. TABELAS DE TESTES

I.1 Teste 1

Teste feito com um conjunto de dados de tamanho 500 e com um vetor de entrada de tamanho 20.

I.2 Teste 2

Teste feito com um conjunto de dados de tamanho 1000 e com um vetor de entrada de tamanho 20.

I.3 Teste 3

Teste feito com um conjunto de dados de tamanho 2000 e com um vetor de entrada de tamanho 20.

I.4 Teste 4

Teste feito com um conjunto de dados de tamanho 500 e com um vetor de entrada de tamanho 40.

I.5 Teste 5

Teste feito com um conjunto de dados de tamanho 1000 e com um vetor de entrada de tamanho 40.

I.6 Teste 6

Teste feito com um conjunto de dados de tamanho 2000 e com um vetor de entrada de tamanho 40.

I.7 Projeto da aplicação no SVN

Para baixar a aplicação, utilize um aplicativo com suporte ao controle de versão SVN e digite:
svn checkout <http://tgcaius.googlecode.com/svn/trunk/tgcaius-read-only>

Tabela I.1: Teste 1 - 2 Neurônios

Neurônios: 2		
Letra	Acerto(%)	Acerto Outros(%)
A	59.20	89.93
B	82.25	99.77
C	86.15	94.16
D	52.50	97.08
H	94.90	99.80
I	86.70	94.29
L	96.25	99.62
V	61.35	95.34
W	66.20	98.98
Y	100.00	92.72

Tabela I.2: Teste 1 - 3 Neurônios

Neurônios: 3		
Letra	Acerto(%)	Acerto Outros(%)
A	63.45	91.83
B	90.90	99.89
C	88.60	96.56
D	14.85	66.32
H	91.20	99.57
I	82.35	99.26
L	91.80	98.96
V	73.90	93.55
W	82.30	96.49
Y	87.25	92.42

Tabela I.3: Teste 1 - 4 Neurônios

Neurônios: 4		
Letra	Acerto(%)	Acerto Outros(%)
A	66.50	95.12
B	78.90	100.00
C	12.65	96.21
D	54.20	99.66
H	89.15	98.84
I	80.45	98.26
L	92.55	98.97
V	81.45	86.99
W	77.85	98.61
Y	100.00	93.41

Tabela I.4: Teste 1 - 5 Neurônios

Neurônios: 5		
Letra	Acerto(%)	Acerto Outros(%)
A	55.45	97.80
B	89.10	99.95
C	93.40	93.59
D	60.25	98.25
H	91.80	99.70
I	83.80	95.22
L	82.10	99.48
V	83.70	96.63
W	74.40	96.58
Y	100.00	92.04

Tabela I.5: Teste 1 - 6 Neurônios

Neurônios: 6		
Letra	Acerto(%)	Acerto Outros(%)
A	43.60	96.04
B	45.20	100.00
C	91.90	98.97
D	63.65	97.27
H	87.40	99.93
I	97.20	97.83
L	73.10	99.81
V	58.10	98.64
W	74.95	93.99
Y	93.95	95.63

Tabela I.6: Teste 1 - 7 Neurônios

Neurônios: 7		
Letra	Acerto(%)	Acerto Outros(%)
A	30.60	95.05
B	81.55	96.54
C	87.65	95.29
D	56.45	99.31
H	91.85	99.07
I	95.05	96.06
L	89.30	99.33
V	87.25	97.71
W	70.25	95.19
Y	100.00	89.49

Tabela I.7: Teste 1 - 8 Neurônios

Neurônios: 8		
Letra	Acerto(%)	Acerto Outros(%)
A	50.90	95.42
B	85.40	100.00
C	96.25	94.59
D	51.90	99.56
H	77.10	99.95
I	94.10	96.90
L	57.20	98.64
V	78.80	97.38
W	75.15	97.74
Y	99.90	93.16

Tabela I.8: Teste 1 - 9 Neurônios

Neurônios: 9		
Letra	Acerto(%)	Acerto Outros(%)
A	55.00	93.42
B	56.85	99.65
C	94.65	98.74
D	50.40	97.75
H	94.50	99.54
I	99.50	96.57
L	90.55	99.88
V	68.20	97.26
W	77.60	96.20
Y	100.00	94.07

Tabela I.9: Teste 2 - 2 Neurônios

Neurônios: 2		
Letra	Acerto(%)	Acerto Outros(%)
A	37.35	98.93
B	65.45	100.00
C	98.55	98.08
D	6.70	99.24
H	85.00	97.67
I	68.95	94.96
L	98.30	99.88
V	79.80	95.11
W	63.65	99.79
Y	99.85	92.06

Tabela I.10: Teste 2 - 3 Neurônios

Neurônios: 3		
Letra	Acerto(%)	Acerto Outros(%)
A	37.95	99.51
B	82.60	100.00
C	98.00	98.11
D	12.80	99.43
H	90.40	99.95
I	68.55	94.92
L	95.40	99.72
V	79.20	96.09
W	45.45	99.97
Y	100.00	88.01

Tabela I.11: Teste 2 - 4 Neurônios

Neurônios: 4		
Letra	Acerto(%)	Acerto Outros(%)
A	31.20	98.72
B	88.60	99.39
C	99.90	98.30
D	20.70	99.96
H	94.20	99.93
I	67.00	95.62
L	94.80	99.08
V	56.30	93.38
W	58.00	98.07
Y	99.65	89.07

Tabela I.12: Teste 2 - 5 Neurônios

Neurônios: 5		
Letra	Acerto(%)	Acerto Outros(%)
A	31.15	99.21
B	93.65	99.99
C	99.80	99.18
D	22.75	99.93
H	94.60	99.31
I	75.85	93.29
L	96.20	99.96
V	88.35	95.95
W	38.60	99.61
Y	99.80	89.62

Tabela I.13: Teste 2 - 6 Neurônios

Neurônios: 6		
Letra	Acerto(%)	Acerto Outros(%)
A	35.20	96.53
B	77.75	100.00
C	98.20	99.12
D	10.30	97.93
H	91.30	99.92
I	74.40	95.46
L	97.40	99.46
V	72.70	95.07
W	64.45	99.61
Y	100.00	88.99

Tabela I.14: Teste 2 - 7 Neurônios

Neurônios: 7		
Letra	Acerto(%)	Acerto Outros(%)
A	38.35	99.85
B	90.30	100.00
C	98.95	98.67
D	25.50	99.98
H	90.75	99.87
I	73.15	98.27
L	98.10	99.05
V	78.65	96.63
W	67.70	99.65
Y	99.65	92.77

Tabela I.15: Teste 2 - 8 Neurônios

Neurônios: 8		
Letra	Acerto(%)	Acerto Outros(%)
A	40.35	99.64
B	91.00	100.00
C	98.00	98.23
D	20.65	99.98
H	94.80	99.97
I	71.60	97.05
L	88.30	99.49
V	93.95	96.49
W	63.85	98.97
Y	99.95	89.49

Tabela I.16: Teste 2 - 9 Neurônios

Neurônios: 9		
Letra	Acerto(%)	Acerto Outros(%)
A	33.95	98.62
B	75.85	100.00
C	100.00	98.58
D	34.90	99.67
H	92.70	99.98
I	76.20	95.83
L	94.65	99.77
V	55.20	94.95
W	44.20	99.91
Y	99.85	92.18

Tabela I.17: Teste 3 - 2 Neurônios

Neurônios: 2		
Letra	Acerto(%)	Acerto Outros(%)
A	62.70	99.40
B	98.55	100.00
C	99.05	99.17
D	41.75	99.34
H	96.85	99.86
I	92.40	98.22
L	93.30	99.45
V	92.60	95.18
W	59.30	98.85
Y	92.90	97.73

Tabela I.18: Teste 3 - 3 Neurônios

Neurônios: 3		
Letra	Acerto(%)	Acerto Outros(%)
A	52.15	99.81
B	96.45	99.98
C	99.95	99.93
D	72.45	97.78
H	97.55	99.77
I	78.35	99.09
L	93.50	99.24
V	97.30	95.91
W	51.50	98.18
Y	97.85	97.57

Tabela I.19: Teste 3 - 4 Neurônios

Neurônios: 4		
Letra	Acerto(%)	Acerto Outros(%)
A	65.55	97.28
B	97.35	99.97
C	99.70	99.76
D	64.00	97.69
H	92.65	99.89
I	82.00	97.32
L	77.00	99.93
V	93.10	94.10
W	47.20	99.58
Y	99.65	96.06

Tabela I.20: Teste 3 - 5 Neurônios

Neurônios: 5		
Letra	Acerto(%)	Acerto Outros(%)
A	58.60	99.39
B	87.85	100.00
C	99.75	99.66
D	62.55	99.21
H	96.95	99.96
I	87.30	99.52
L	89.80	99.69
V	94.75	93.84
W	59.85	97.95
Y	99.65	96.46

Tabela I.21: Teste 3 - 6 Neurônios

Neurônios: 6		
Letra	Acerto(%)	Acerto Outros(%)
A	70.90	98.18
B	98.80	100.00
C	99.95	99.60
D	82.70	98.70
H	97.65	99.81
I	94.05	98.06
L	85.65	99.96
V	95.45	94.93
W	69.10	98.64
Y	98.80	97.18

Tabela I.22: Teste 3 - 7 Neurônios

Neurônios: 7		
Letra	Acerto(%)	Acerto Outros(%)
A	60.65	99.32
B	92.75	99.97
C	99.60	99.69
D	59.25	99.47
H	96.35	99.89
I	89.55	99.46
L	91.80	99.66
V	96.75	96.06
W	39.75	99.33
Y	99.50	96.71

Tabela I.23: Teste 3 - 8 Neurônios

Neurônios: 8		
Letra	Acerto(%)	Acerto Outros(%)
A	62.95	99.68
B	88.60	100.00
C	94.60	99.77
D	52.40	99.31
H	94.60	99.92
I	81.35	98.46
L	88.65	99.91
V	92.55	95.38
W	59.05	98.83
Y	99.90	97.38

Tabela I.24: Teste 3 - 9 Neurônios

Neurônios: 9		
Letra	Acerto(%)	Acerto Outros(%)
A	64.80	98.18
B	97.20	99.97
C	99.55	99.92
D	79.35	98.37
H	98.00	99.78
I	88.45	97.59
L	93.05	99.99
V	97.25	92.93
W	55.50	98.49
Y	99.90	98.25

Tabela I.25: Teste 4 - 2 Neurônios

Neurônios: 2		
Letra	Acerto(%)	Acerto Outros(%)
A	96.15	94.83
B	76.50	99.87
C	99.90	95.84
D	40.15	95.81
H	86.40	99.81
I	67.90	99.26
L	75.80	99.99
V	89.95	95.72
W	32.10	98.58
Y	91.30	93.47

Tabela I.26: Teste 4 - 3 Neurônios

Neurônios: 3		
Letra	Acerto(%)	Acerto Outros(%)
A	91.05	96.12
B	99.90	97.88
C	99.00	98.93
D	47.30	96.04
H	95.75	99.79
I	71.95	98.93
L	81.40	99.68
V	91.50	95.59
W	22.80	99.63
Y	86.10	95.07

Tabela I.27: Teste 4 - 4 Neurônios

Neurônios: 4		
Letra	Acerto(%)	Acerto Outros(%)
A	93.25	96.25
B	96.05	97.75
C	98.05	92.40
D	44.50	96.24
H	86.40	99.86
I	83.10	98.19
L	94.80	99.84
V	77.75	94.83
W	26.10	97.71
Y	85.60	89.56

Tabela I.28: Teste 4 - 5 Neurônios

Neurônios: 5		
Letra	Acerto(%)	Acerto Outros(%)
A	90.70	97.21
B	94.80	98.24
C	100.00	98.28
D	45.85	97.33
H	85.75	99.99
I	76.75	98.93
L	61.85	99.86
V	90.10	96.63
W	22.50	98.62
Y	92.20	93.70

Tabela I.29: Teste 4 - 6 Neurônios

Neurônios: 6		
Letra	Acerto(%)	Acerto Outros(%)
A	92.55	95.05
B	97.70	97.68
C	99.15	98.27
D	44.20	97.00
H	65.50	99.82
I	90.75	99.74
L	88.85	99.78
V	89.00	93.71
W	48.85	100.00
Y	84.60	90.76

Tabela I.30: Teste 4 - 7 Neurônios

Neurônios: 7		
Letra	Acerto(%)	Acerto Outros(%)
A	95.90	96.16
B	98.45	97.96
C	100.00	98.39
D	68.20	97.61
H	79.15	99.90
I	86.65	99.82
L	73.80	99.72
V	91.00	95.19
W	28.85	99.87
Y	96.05	95.41

Tabela I.31: Teste 4 - 8 Neurônios

Neurônios: 8		
Letra	Acerto(%)	Acerto Outros(%)
A	90.35	97.83
B	98.30	97.19
C	98.55	97.14
D	64.65	98.09
H	88.00	99.88
I	71.05	99.44
L	70.65	99.62
V	93.15	98.04
W	26.40	99.85
Y	92.60	91.57

Tabela I.32: Teste 4 - 9 Neurônios

Neurônios: 9		
Letra	Acerto(%)	Acerto Outros(%)
A	97.75	95.45
B	98.45	98.92
C	99.45	97.80
D	53.80	97.23
H	75.65	99.98
I	79.15	97.44
L	78.75	99.88
V	96.40	97.63
W	62.35	99.97
Y	86.45	96.41

Tabela I.33: Teste 5 - 2 Neurônios

Neurônios: 2		
Letra	Acerto(%)	Acerto Outros(%)
A	45.05	97.81
B	98.80	97.61
C	99.00	96.02
D	81.25	94.53
H	84.10	99.74
I	62.70	99.25
L	81.90	98.48
V	69.25	96.89
W	73.70	98.01
Y	82.95	94.24

Tabela I.34: Teste 5 - 3 Neurônios

Neurônios: 3		
Letra	Acerto(%)	Acerto Outros(%)
A	90.35	94.64
B	99.35	97.78
C	100.00	98.89
D	75.05	91.52
H	92.80	98.69
I	55.90	99.65
L	81.20	98.49
V	79.80	96.09
W	82.20	98.22
Y	83.70	95.99

Tabela I.35: Teste 5 - 4 Neurônios

Neurônios: 4		
Letra	Acerto(%)	Acerto Outros(%)
A	63.95	99.25
B	95.35	99.54
C	99.85	99.23
D	76.75	94.57
H	92.00	96.77
I	56.35	99.81
L	68.65	98.96
V	75.80	97.80
W	89.75	99.65
Y	81.65	97.66

Tabela I.36: Teste 5 - 5 Neurônios

Neurônios: 5		
Letra	Acerto(%)	Acerto Outros(%)
A	69.65	96.34
B	92.70	98.15
C	99.75	97.46
D	57.65	93.47
H	90.95	98.31
I	68.85	99.69
L	83.00	99.22
V	64.35	98.58
W	90.70	99.23
Y	83.25	96.02

Tabela I.37: Teste 5 - 6 Neurônios

Neurônios: 6		
Letra	Acerto(%)	Acerto Outros(%)
A	81.65	94.51
B	98.20	99.20
C	100.00	96.99
D	83.70	94.31
H	94.40	99.83
I	78.05	99.96
L	85.10	99.41
V	69.75	98.26
W	91.70	98.83
Y	84.30	98.03

Tabela I.38: Teste 5 - 7 Neurônios

Neurônios: 7		
Letra	Acerto(%)	Acerto Outros(%)
A	75.60	98.45
B	99.00	98.60
C	99.65	98.42
D	73.25	93.08
H	84.95	99.88
I	61.10	99.51
L	87.20	98.04
V	75.15	98.82
W	94.70	99.63
Y	84.10	98.17

Tabela I.39: Teste 5 - 8 Neurônios

Neurônios: 8		
Letra	Acerto(%)	Acerto Outros(%)
A	74.00	98.74
B	94.80	99.14
C	100.00	98.64
D	77.85	94.79
H	84.85	99.75
I	50.25	97.96
L	79.25	99.95
V	68.55	99.78
W	88.80	99.58
Y	90.50	96.64

Tabela I.40: Teste 5 - 9 Neurônios

Neurônios: 9		
Letra	Acerto(%)	Acerto Outros(%)
A	94.40	93.56
B	97.55	98.84
C	98.45	99.72
D	81.95	93.43
H	88.95	97.50
I	50.25	99.93
L	71.85	99.63
V	68.50	98.98
W	92.50	99.82
Y	91.25	98.43

Tabela I.41: Teste 6 - 2 Neurônios

Neurônios: 2		
Letra	Acerto(%)	Acerto Outros(%)
A	85.80	96.14
B	100.00	95.79
C	97.70	99.93
D	72.85	99.42
H	93.50	93.85
I	96.25	99.83
L	97.40	96.74
V	72.65	97.44
W	70.05	99.99
Y	88.85	95.54

Tabela I.42: Teste 6 - 3 Neurônios

Neurônios: 3		
Letra	Acerto(%)	Acerto Outros(%)
A	71.70	91.27
B	100.00	97.44
C	91.75	99.92
D	59.75	98.99
H	77.60	99.76
I	87.50	99.59
L	96.70	92.36
V	80.60	97.16
W	79.90	99.49
Y	93.35	95.54

Tabela I.43: Teste 6 - 4 Neurônios

Neurônios: 4		
Letra	Acerto(%)	Acerto Outros(%)
A	85.35	97.04
B	99.45	98.31
C	100.00	99.99
D	62.05	98.42
H	61.80	97.97
I	94.85	99.50
L	96.15	96.59
V	63.55	99.29
W	72.40	99.24
Y	94.85	97.34

Tabela I.44: Teste 6 - 5 Neurônios

Neurônios: 5		
Letra	Acerto(%)	Acerto Outros(%)
A	74.80	95.80
B	100.00	98.39
C	98.00	99.79
D	76.95	99.11
H	82.20	99.06
I	90.15	99.09
L	98.70	96.86
V	71.15	98.34
W	76.75	99.76
Y	92.90	97.19

Tabela I.45: Teste 6 - 6 Neurônios

Neurônios: 6		
Letra	Acerto(%)	Acerto Outros(%)
A	92.15	97.96
B	99.90	97.51
C	100.00	99.97
D	78.15	99.02
H	87.25	99.79
I	92.50	99.63
L	97.70	99.19
V	80.15	99.37
W	65.70	99.98
Y	93.00	93.44

Tabela I.46: Teste 6 - 7 Neurônios

Neurônios: 7		
Letra	Acerto(%)	Acerto Outros(%)
A	96.35	97.25
B	98.95	98.21
C	98.15	99.90
D	72.15	98.63
H	84.25	100.00
I	92.35	99.21
L	96.80	99.78
V	76.35	97.24
W	70.80	99.92
Y	92.50	95.03

Tabela I.47: Teste 6 - 8 Neurônios

Neurônios: 8		
Letra	Acerto(%)	Acerto Outros(%)
A	97.25	97.14
B	100.00	96.19
C	96.95	100.00
D	71.05	98.83
H	73.60	99.82
I	85.75	99.67
L	99.40	99.36
V	83.05	99.07
W	72.35	99.63
Y	94.85	96.58

Tabela I.48: Teste 6 - 9 Neurônios

Neurônios: 9		
Letra	Acerto(%)	Acerto Outros(%)
A	99.00	98.04
B	99.80	98.31
C	96.30	99.82
D	56.35	99.31
H	86.85	99.87
I	92.45	99.61
L	97.75	99.74
V	67.20	99.86
W	72.95	99.47
Y	94.10	92.54

II. DESCRIÇÃO DO CONTEÚDO DO CD

No CD existem 3 pastas numeradas e com o seguinte conteúdo:

1. Monografia no formato PDF.
2. Programa Eclipse, arquivos binários com redes neurais previamente treinadas utilizadas nos resultados experimentais e todo o código da aplicação em C com o projeto no Eclipse e o código utilizado para o programa matlab nos formatos originais e no formato PDF.
3. Resumo do TG e palavras-chave.